SilverStorm 9000
CLI Reference Guide

D000025-002 Rev. A

Information furnished in this manual is believed to be accurate and reliable. However, QLogic Corporation assumes no responsibility for its use, nor for any infringements of patents or other rights of third parties which may result from its use. QLogic Corporation reserves the right to change product specifications at any time without notice. Applications described in this document for any of these products are for illustrative purposes only. QLogic Corporation makes no representation nor warranty that such applications are suitable for the specified use without further testing or modification. QLogic Corporation assumes no responsibility for any errors that may appear in this document.

Document Revision History	
Revision A, August 30, 2008	
Changes	**Sections Affected**

© 2008 QLogic Corporation. All Rights Reserved Worldwide.
First Published: September 2007

QLogic Corporation, 26650 Aliso Viejo Parkway, Aliso Viejo, CA 92656, (800) 662-4471 or (949) 389-6000

Table of Contents

1 **Introduction**

Intended Audience 1-1
License Agreements 1-1
Technical Support 1-2
Availability 1-2
Contact Information 1-2

2 **Switch Command Line Interface**

Overview 2-1
Commands and Functional Groups 2-2
Online Help 2-4
Keyboard Shortcuts 2-4
Accessing the CLI 2-4
Groups and Commands 2-5
General 2-5
Chassis 2-28
Network 2-33
Firmware 2-39
Subnet Management 2-47
Log 2-74
Key Management 2-80
IbSwitchInfo 2-82
TimeManagement 2-99
SNMP 2-103
Capture 2-109

3 **Virtual I/O Command Line Interface**

Overview 3-1
Commands and Functional Groups 3-1
Online Help 3-3
Keyboard Shortcuts 3-3
Accessing the VIO CLI 3-3
Groups and Commands 3-4
General 3-4

Linecard 3-4
Firmware 3-13
Log 3-16
Capture 3-16
FibreChannel 3-19
Ethernet 3-45
Configuration 3-88

1 Introduction

This manual describes the command line interface (CLI) task information for the SilverStorm™ 9024 and Silverstom Multi-Protocol Fabric Director (MPFD) family of switches.

This manual is organized as follows:

Section 1 describes the intended audience and technical support.

Section 2 describes the 9000 switch CLI commands.

Section 3 describes all the Virtual I/O (EVIC and FVIC) CLI commands.

Intended Audience

This manual is intended to provide network administrators and other qualified personnel a reference for the command line interface of the SilverStorm 9000 series switches.

License Agreements

Refer to the *QLogic Software End User License Agreement* for a complete listing of all license agreements affecting this product.

Technical Support

Customers should contact their authorized maintenance provider for technical support of their QLogic switch products. QLogic-direct customers may contact QLogic Technical Support; others will be redirected to their authorized maintenance provider.

Visit the QLogic support Web site listed in Contact Information for the latest firmware and software updates.

Availability

QLogic Technical Support for products under warranty is available during local standard working hours excluding QLogic Observed Holidays.

Contact Information

Support Headquarters	QLogic Corporation 4601 Dean Lakes Blvd. Shakopee, MN 55379 USA
QLogic Web Site	www.qlogic.com
Technical Support Web Site	support.qlogic.com
Technical Support Email	support@qlogic.com
Technical Training Email	tech.training@qlogic.com
North American Region	
Email	support@qlogic.com
Phone	+1-952-932-4040
Fax	+1 952-687-2504
All other regions of the world	
QLogic Support Web Site	www.support.qlogic.com

2 Switch Command Line Interface

Overview

This section details the usage of the Command Line Interface (CLI) feature for the SilverStorm 9000 family of switches:

- The SilverStorm 9024
- The SilverStorm 9000 Multi-protocol Fabric Director (MPFD) Series:
  - SilverStorm 9020
  - SilverStorm 9040
  - SilverStorm 9080
  - SilverStorm 9120
  - SilverStorm 9240

The CLI allows the user to perform remote configuration and management tasks, which in many respects mirrors the functionality of the Chassis Viewer GUI.

The CLI is accessed via a terminal attached to the switch spine module(s) RS232 port(s) or via the OOB management port using Telnet and secure shell (SSH). For a standalone switch, the user would Telnet to the IP address(es) of the unit. Once connected, the CLI works as any telnet session does.

To access the CLI, a login and password is required. There are two user modes, operator and administrator with the following access privileges:

Operator:

- Read only access.

Administrator:

- Read and write Access.
- Reboot access.
- Can change operator and administrator passwords.

- Can disable user login and passwords. This would allow all users admin-level access without the need for a user name or password.
- Can view all current user sessions
- Can access all of the commands executed from any open operator session.
- Can log out any open user sessions
- Can send messages to the open user sessions

The CLI allows multiple users to be logged in simultaneously. However, some commands will be locked to a user(s) if another user is executing the same command.

The CLI is also accessible through the RS232 serial port of the switch. Accessing the CLI through the serial port never requires a login and password and defaults to administrator privileges. Providing access through the serial port means that users will always have access to the switch, even if Telnet and SSH are not functioning.

Commands and Functional Groups

The list of available commands can be accessed by typing **list**. To keep the list short, the commands are grouped into functional groups, which are:

General:

General commands for user management and CLI configuration.

Deprecated:

Commands that have been deprecated.

The Deprecated group contains CLI commands that have been replaced or are to be removed. Please use the new command where appropriate.

Chassis:

Chassis management commands. (FRU info, fan/power supply state, etc).

Network:

Ethernet interface management commands.

Firmware:

Provides commands for updating the firmware via a File Transfer Protocol (FTP) server or Secure Copy Protocol (SCP) (if using SSH to access the Bridge Module). The switch has the ability to store the location of the firmware files for future upgrades. Additionally, the Firmware functional group includes commands for viewing the current firmware revisions and for changing the boot image.

SubnetManagement:

InfiniBand subnet manager configuration and management.

Log:

Provides commands for viewing log files as well as configuring logging parameters.

KeyManagement:

License key management.

IbSwitchInfo:

Provides commands for displaying InfiniBand (IB) statistics for all IB ports on the switch, as well as for configuring port statistic thresholds.

TimeManagement:

Provides commands for retrieving and setting the current system time, as well as commands for setting the time zone and daylight saving time parameters.

Snmp:

Provides commands for configuring SNMP trap destinations and security parameters required to access the switch from an SNMP manager.

Capture:

Provides commands for capturing switch-specific information for the purposes of analysis and debugging.

To list commands within a functional group, simply type in the functional group name. For example, to list all of the firmware commands, type **list Firmware**. The system would display the following:

```
-> list Firmware
fwUpdateSlot         Update units firmware
fwUpdateChassis      Update units firmware
fwListFiles          List the contents of the firmware ramdisk
fwShowUpdateParams   Display firmware default update parameters
fwSetUpdateParams    Configure firmware default update parameters
showCapability       Display the capabilities/features
showLastScpRetCode  Display the return code from the last SCP Firmware Push
fwVersion            Display Firmware revisions
bootQuery            Query boot image information
bootSelect           Change boot selection
```

Online Help

The online help for the CLI provides, for each command, all necessary information to successfully execute the command. For example, typing **help list** (can also type **list help**) displays the following information for the `list` command:

```
NAME
   list
SYNOPSIS
   list [group] [-noprompt] [-verbose]
DESCRIPTION
   List available commands.
OPTIONS
   group     - List the commands in that particular group
   -noprompt - Just list the command groups.
   -verbose  - Print full help for each command, instead of summary.
NOTES
   Specify the group name or use 'all' to list all available commands.
```

Keyboard Shortcuts

- The CLI keeps a history of recently executed commands. This history is available via the **Up** and **Down** arrow keys.
- Users may edit the current command with the **Left** and **Right** arrow keys.
- Tab completion: pressing the **Tab** key after typing at least one character either completes a command or lists all the available commands that begin with the characters already typed.

Accessing the CLI

NOTE: The CLI can be accessed via Telnet, SSH, or through the switch RS232 serial port. The following instructions use Telnet.

1. Telnet to the IP address of the switch (the default IP address is 192.168.100.9) with the following command:

   telnet <IP ADDRESS>

2. The system prompts for a username. The CLI has the following default user names:

   Operator access: **operator**

   Administrator access: **admin**

   Type the appropriate username and press **Enter**.

3. The system prompts for a password. The CLI has the following default passwords:

   Operator access: **operpass**

   Administrator access: **adminpass**

   Type the appropriate password and press **Enter**. The system responds with:

   ```
   Welcome to the <SWITCH> CLI. Type 'list' for the list of
   commands.
   ```

Groups and Commands

The following section lists all CLI functional groups along with the commands for each group. Commands for all 9000 switches are listed. Any commands specific to a switch(es) is noted. For more specific information for each functional group, the user would execute the **help <GROUP NAME>** command. For more specific command information, the user would execute the **help <COMMAND NAME>** command.

General

help

Displays help information for a specific command.

Syntax:

```
help [command]
```

Options:

[command]

The command to display help for.

Sample Output:

```
-> help list

NAME
       list

SYNOPSIS
       list [group] [-noprompt]

DESCRIPTION
       List available commands.

OPTIONS
       group     - List the commands in that particular group

       -noprompt - Just list the command groups.
```

NOTES: General Help

Type list or ? for the list of commands.

To get help on a particular command type: commandname help.

For convienence purposes you can also type: help commandname

Use the Up and Down arrow keys to browse command history, Left and Right arrow keys to edit the current command and the Tab key for tab completion of a command.

Two alternate key bindings exist for the backspace and delete keys. If these keys are not responding as expected use the **swapBsDel** command to swap the bindings.

Commands are grouped into subcategories. To list the commands in a subcategory type in the category heading. Category headings are identified by starting with a capital letter. For example, to list all the commands that handle log configuration type Log.

list

Displays a list of all valid commands.

Syntax:

```
list [group] [-noprompt] [-verbose]
```

Options:

group

Displays a list of commands for a particular group

-noprompt

Displays a list of the command groups only.

-verbose

Print full help for each command, instead of summary.

Sample Output:

```
list
List of Valid Commands:
General         General commands for user management and CLI configuration.
Deprecated      These commands have been deprecated
Network         Snmp configuration commands.
Firmware        Update firmware and display current revision levels.
Log             Log file display and configuration
IbSwitchInfo    InfiniBand port configuration and statistics
TimeManagement Display and configure the system time
Snmp            Snmp configuration commands.
Ethernet        Provides commands for managing the Ethernet interface.

Type the name of the group you want to list or return to exit:
```

NOTES: Specify the group name or use the option **all** to list all available commands. To get a full list of commands and all help text, user **list all -verbose**.

history

Displays the command history for this session.

Syntax:

```
history
```

Options:

None.

Sample Output:

```
-> history
command history [30 max lines]:
  list
  Deprecated
  Network
  Firmware
  Log
  IbSwitchInfo
  TimeManagement
  Snmp
  Ethernet
  help
  list
  history
```

reboot

Reboots the device.

Syntax:

```
reboot [now][-m][slot n][-s][-n][all]
```

Options:

now

Does not prompt the user before rebooting.

-m

Reboot Master (local) Spine 1 (non-disruptive).

slot n

reset specific device where n = slotNumber (disruptive).

-s

Reboot Slave (remote) Spine 2.

-n

Reboot Slave (remote) management card only (non-disruptive)

all

Reboot All local devices (excludes -n/-m/-s).

Sample Output:

```
-> reboot
Disruptive reboot selected
Proceed with reboot? [N]
```

NOTES: You may reboot the local (Master) Spine or the remote (Slave) Spine with one or multiple arguments.

Non-disruptive arguments will not interfere with switch traffic.

Providing the argument **all** performs disruptive reboot of all present Spines and cards.

If rebooting the local device from telnet, ssh, etc., you will have to reconnect after rebooting.

Default (no arguments) reboots local device disruptively after prompt.

See also: 'resetCard', 'showInventory'.

killCliSession

Terminates an existing CLI session

Syntax:

```
killCliSession sessionNumber
```

Options:

sessionNumber

The sessionNumber that is returned from the **who** command.

Sample Output:

```
-> killCliSession
must supply session number
```

NOTES: This command logs out remote sessions. Use 'who' to obtain the list of active sessions.

who

Displays currently active CLI sessions.

Syntax:

```
who
```

Options:

None.

Sample Output:

```
user                role       index   logged in             last cmd              type    ip address
-------------------------------------------------------------------------------------------------
admin               admin      0       13:19:03 07/11/2008   09:41:20 07/11/2008   serial  0.0.0.0
```

NOTES: Displays a list of currently active CLI sessions. Note that a session can be 'active', but no user information available (in most cases, this indicates the session is waiting for the user to enter login information).

For each session the following information is displayed:

- user: username of the logged in user
- role: security role of the user
- index: internal session index
- logged in: timestamp of when the user logged in
- last cmd: timestamp of the users last command
- type: method used to connect to the system
- ip address: ip address of the user (if applicable)

broadcast

Write a message to all active CLI sessions

Syntax:

```
broadcast msg
```

Options:

msg
Message (Note: the message text must be encapsulated in quotes "".

Sample Output:

```
-> broadcast "The system will be rebooted in 5 minutes."
```

NOTES: Writes the supplied message to all other active CLI sessions. The message to be written must be encapsulated in quotes, and non-empty.

swapBsDel

Swaps the backspace and delete character key bindings.

Syntax:

```
swapBsDel
```

Options:

None.

Sample Output:

```
-> swapBsDel
```

NOTES: Terminals may bind the backspace and delete key bindings differently. This command swaps two commonly used bindings, which allow the user to use the backspace and delete keys properly without having to adjust their terminal settings. Backspace and delete swapping is persistently maintained per user (i.e. each login account can have a separate binding).

setTermWidth

Change the terminal width for text formatting purposes.

Syntax:

```
setTermWidth width
```

Options:

width
Width of the user's terminal window. Minimum width is 20 characters.

Sample Output:

```
-> setTermWidth 100
```

NOTES: Allows modification of the terminal width used for text formatting purposes. Note that not all commands adhere to this setting. The minimum width is 20 characters.

getTermWidth

Displays the terminal width for text formatting purposes.

Syntax:

```
getTermWidth
```

Options:

None.

Sample Output:

```
-> getTermWidth
Current terminal width: 80 characters.
```

NOTES: Displays the terminal width used for text formatting purposes. Note that not all commands adhere to this setting.

prompt

Set the CLI prompt (global for all active CLI sessions).

Syntax:

```
prompt str
```

Options:

str
The new prompt.

Sample Output:

```
prompt -> "9024->"
9024->
```

NOTES: This changes the prompt for all CLI sessions. The prompt may not exceed 11 characters and is not saved across reboots. If the prompt contains a space, asterick, comma, parenthesis or semicolon it must be enclosed with double quotes ". For example: "*a prompt*". Also, if a prompt is not accepted try to enclose it with double quotes.

case

Display or change the case sensitivity for the command interpreter for the CLI session.

Syntax:

```
case [off | on]
```

Options:

off | on
Turn case sensitivity off or on.

Sample Output:

```
-> case off
Case sensitivity is now off
```

NOTES: This displays or changes the case sensitivity for the command interpreter for the CLI session. If an option is not specified the current case sensitivity is displayed, otherwise the case sensitivity is turned on or off depending on the specified option. When case sensitivity is on, the CLI input must match the exact character case (lower and upper case) as specified in the help text. When case sensitivity is turned off, the input may be any combination of upper and lower case.

showLastRetCode

Display the return code from the last executed command.

Syntax:

```
showLastRetCode [-brief]
```

Options:

-brief
Displays just the numeric value of the return code.

Sample Output:

```
-> showLastRetCode
Last Exit Code: 0: Success
```

NOTES: This allows for automated systems to determine if a command was successful or not.

echo

Echoes text back to the output..

Syntax:

```
echo [<text> ...]
```

Options:

```
[<text> ...]
```
Text to be echoed..

Sample Output:

```
MasterSpine1-> echo test
test
```

NOTES: This echoes the text for each argument back to the output. If there are multiple arguments they are separated by spaces.

rlogin

Creates I/O terminal to local cards.

Syntax:

```
rlogin [hostName]
```

Options:

```
hostName
```
The name of the device to connect to.

Sample Output:

```
-> rlogin slot2
```

NOTES: This command allows users to open a terminal to local I/O devices within the chassis/hemisphere. The command rcmd is an alias of rlogin. See also hostShow.

rcmd

Creates I/O terminal to local cards.

Syntax:

```
rcmd [hostName]
```

Options:

hostName
The name of the device to connect.

Sample Output:

```
-> rcmd Host1
```

NOTES: This command allows users to open a terminal to local I/O devices within the chassis/hemisphere. The command rcmd is an alias of rlogin. See also hostShow.

resetCard

Reboot a specific slot in the chassis.

Syntax:

```
resetCard n [now]
```

Options:

n
Chassis slot number.

now
No prompt before resetting the card.

Sample Output:

```
-> resetCard 2 now
```

NOTES: If you reset the switch you are connected to you will have to reconnect. You can find out the valid slots for this device with the 'showInventory' command.

logout

Logout of the current CLI session.

Syntax:

```
logout
```

Options:

None.

Sample Output:

```
logout
```

user

Change user accounts.

Syntax:

```
user [user]
```

Options:

[user]
Name of user account to change to.

Sample Output:

```
-> user operator
User changed to: operator
```

NOTES: Use this command to change to the 'operator' account, or to the 'admin' account.

passwd

Change users password(s).

Syntax:

```
passwd [username]
```

Options:

[username]
Name of the user whose password needs to be changed.

Sample Output:

```
-> passwd operator1
User password changed successfully
```

NOTES: Allows the modification of the user's password.

userAdd

Add a user account.

Syntax:

```
userAdd [role] [username] [password]
```

Options:

[role]

Can be either **admin** or **operator**.

[username]

The name of the user to be added.

[password]

Optional password. If not supplied the default password for that role is used.

Sample Output:

```
-> userAdd admin Bob
User added: Bob
Password is set to the default password for this role: admin
```

userRem

Remove a user account.

Syntax:

```
userRem [username]
```

Options:

username

Name of the user to be removed.

Sample Output:

```
-> userRem Bob
User deleted: Bob
```

userListShow

List all user accounts for this device.

Syntax:

```
userListShow
```

Options:

None.

Sample Output:

```
-> userListShow
username              role
operator1             operator
USERID                admin
Bob                   admin
```

sshKey

Syntax:

```
sshKey <cmd> [-u <username>] [<input>]
```

Options:

show

Displays the SSH public keys in the users authorized_keys file.

add "<key>"

Add key to the users authorized_keys file. Must be enclosed within double-quotes ("<key>").

rem <index>

Remove key at <index> for the user.

rem -all

Remove all keys for the usern.

-u <username>

Perform the operation on the user <username> (for admins only).

Sample Output:

```
-> sshKey show
 Index   Key
-----------------------------------
   1     "ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEArNiSexu30rZjs1HAXbDBwTgJgcxLF..."
```

NOTES: Adds, displays, and removes SSH public keys for individual users. Users with administrative privileges may use the -u option to manage keys for other users.

loginMode

Allows the user to change the user interface authentication requirements.

Syntax:

```
loginMode [mode]
```

Options:

```
[mode]
```

Determines how users are able to login.

Sample Output:

```
-> loginMode 1
Mode successfully changed to: 1 = Password is not required
```

NOTES: This command displays or changes how users are authenticated when connecting to the GUI or CLI. With no parameters, the current login mode is displayed. The login mode can be changed by specifying a single integer parameter indicating which of the following modes should become active:

0 - Username and password required

1 - Password is not required

2 - Username / password are not required

3 - LDAP Authentication

When usernames are disabled, all users will be logged on as the administrative user. There is no way to change this behavior.

setldapSvrIpAddr

Modify the LDAP Server IP Address.

Syntax:

```
setldapSvrIpAddr [ipaddress]
```

Options:

```
[ipaddress]
```

The IP address of the ldap server you wish to set to authenticate in the format "192.168.0.1"

Sample Output:

```
-> setldapSvrIpAddr 192.168.0.29
```

NOTES: This sets the ldap server ip address. The ldap server will be contacted for remote authentication.

setldapSvrPort

Modify the LDAP Server Port.

Syntax:

```
setldapSvrPort [port]
```

Options:

[port]
The port of the LDAP server to be authenticated.

Sample Output:

```
-> setldapSvrPort 389
```

NOTES: Sets the LDAP server port. The LDAP server will be contacted for remote authentication.

idleTimeoutGet

Retrieves the UI idle timeout value.

Syntax:

```
idleTimeoutGet [--all] [--cli] [--gui]
```

Options:

--all
Display all timeouts. This is the default if no options are specified.

--cli
Display the timeout for CLI sessions.

--gui
Display the timeout for GUI sessions.

Sample Output:

```
-> idleTimeoutGet
Timeout is 600 seconds.
```

NOTES: Displays the system idle timeouts for the GUI and CLI interfaces. If set to zero, the timeout is disabled.

idleTimeoutSet

Set the UI idle timeout value.

Syntax:

```
idleTimeoutSet [--all | --cli | --gui] seconds]
```

Options:

[seconds]

Idle timeout value (in seconds).

--all

Set the idle timeout for both the CLI and the GUI to the same value.

--cli

Set the idle timeout for the CLI. This is the default if no identifier is specified.

--gui

Set the idle timeout for the GUI.

Sample Output:

```
-> idleTimeoutSet --all 700
Timeout is set to 700 seconds.
```

NOTES: Modifies the idle timeout for the CLI or GUI interfaces. Timeouts are in number of seconds, with zero meaning the timeout is disabled.

sessionTimeoutDisable

Disable the idle timeout for the current CLI session.

Syntax:

```
sessionTimeoutDisable
```

Options:

None.

Sample Output:

```
-> sessionTimeoutDisable
Disabled session idle timeout.
```

NOTES: Disables the idle timeout for the current CLI session. This value does not persist across instances of the session (i.e., each time you log on, it will default back to the system default value).

sessionTimeoutEnable

Enable the idle timeout for the CLI session.

Syntax:

```
sessionTimeoutEnable
```

Options:

None.

Sample Output:

```
-> sessionTimeoutEnable
Enabled session idle timeout.
```

NOTES: Enables the idle timeout for the current CLI session. This value does not persist across instances of the session (i.e., with each log on, it defaults back to the system default value).

loginMsgGet

Displays the CLI login message for this device.

Syntax:

```
loginMsgGet
```

Options:

None.

Sample Output:

```
-> loginMsgGet
Welcome message: Be certain to logout when you are finished using the
CLI.
```

NOTES: Displays the current login welcome message for the CLI. This message can be customized with the loginMsgSet command.

loginMsgSet

Set the CLI login message for this device.

Syntax:

```
loginMsgSet [message]
```

Options:

[message] The login message (should be encapsulated in quotes “ “).

Sample Output:

```
-> loginMsgSet "Be certain to logout when you are finished using the
CLI."
Welcome message set successfully
```

NOTES: This command can be used to modify the welcome message displayed when logging onto the CLI. The current message can be viewed with the loginMsgGet command.

loginNameGet

Display the name of the device displayed when logged into via telnet.

Syntax:

```
loginNameGet
```

Options:

None.

Sample Output:

```
-> loginNameGet
login-name: Switch1
```

NOTE: The login name is an arbitrary string displayed prior to a user attempting a login to a new CLI session. This command displays the current value of this string.

loginNameSet

Modify the string displayed to users prior to attempting a login for a new CLI session.

Syntax:

```
loginNameSet <text string>
```

Options:

None.

Sample Output:

```
-> loginNameSet Switch1
login-name set successfully
```

NOTES: The login name is an arbitrary string displayed prior to a user attempting a login to a new CLI session. This command modifies this string.

serialAuthGet

Displays the current serial authentication status.

Syntax:

```
serialAuthGet
```

Options:

None.

Sample Output:

```
-> serialAuthGet
Serial authentication is currently disabled.
```

NOTES: Displays whether user login and authentication is required on the serial console of the system.

serialAuthSet

Enable or disable authentication on the systems serial console.

Syntax:

```
serialAuthSet [enable]
```

Options:

[enable]

Boolean value to enable or disable serial authentication.

Sample Output:

```
-> serialAuthSet 1
Serial authentication set to enabled.
```

NOTES: This command is used to enable or disable user authentication on the serial console. Use 1 as the only parameter to enable authentication, or 0 to disable authentication.

uiConfig

View or configure the user interfaces to this device.

Syntax:

```
uiConfig [-telnet 0|1] [-https 0|1] [-http 0|1] [-ftp
0|1] [-snmp 0|1]
```

Options:

[telnet]

Enable/disable access to the device via telnet..

[https]

Enable or disable https access.

[http]

Enable or disable http access.

[ftp]

Enable or disable ftp access.

[snmp]

Enable or disable SNMP access.

Sample Output:

```
-> uiconfig -telnet 0
Successfully modified configuration.
 Option         Value
-----------------------
 telnet          0
 http            1
 https           0
 ftp             1
 snmp            1
```

NOTES: There are several access methods for this device. With no arguments, this command displays which access methods are available, and whether each method is enabled or disabled.

This command can also be used to enable or disable various access methods. Use '-<proto> 0' to disable a protocol, and '-<proto> 1' to enable. Multiple operations may be specified in a single command.

A reboot is not required for this command. Depending on the specific method, it may take 5-10 seconds for the changes. Note that existing connections (for example a telnet session) will not be affected by disabling the underlying access method.

Any access method supported by the device (but not listed here) cannot be configured and is always enabled.

IpoIbConfigShow (9000 MPFD only)

Displays the IpoIb Enable/Disable setting.

Syntax:

```
IpoIbConfigShow
```

Options:

None.

Sample Output:

```
-> IpoIbConfigShow
```

IpoIbConfigEnable (9000 MPFD only)

Enable IpoIb feature.

Syntax:

```
IpoIbConfigEnable <- IpoIb enable>
```

Options:

```
[- IpoIb enable]
```

Sample Output:

```
-> IpoIbConfigEnable - IpoIb enable
```

NOTES: Only allowed trasition at run time is from disable to enable, by default it is disabled.

IpoIbAddressShow (9000 MPFD only)

Displays the IpoIb IP address and associated netmask.

Syntax:

```
IpoIbAddressShow
```

Options:

None.

Sample Output:

```
-> IpoIbAddressShow
```

IpoIbAddressSet (9000 MPFD only)

Change the IpoIb IP address and associated network mask.

Syntax:

```
IpoIbAddressSet -h ipaddress -m netMask
```

Options:

[-h ipaddress]

the new IP address in dotted notation format 'xxx.xxx.xxx.xxx'.

[-m netMask]

network mask. (in hexidecimal format).

Sample Output:

```
-> IpoIbAddressSet -h 123.45.6.789
```

exit

Exit the CLI.

Syntax:

```
exit
```

Options:

None.

Sample Output:

```
exit
```

Chassis

hwCheck

Runs a system check at the chassis level and returns current hardware status.

Syntax:

```
hwCheck
```

Options:

```
[n - 0=default, 1=verbose]
```

Sample Output:

```
MasterSpine1-> hwCheck
Chassis hardware status: GOOD
```

NOTES: Returns OK or provides detailed status/warning/error information. If an error/warning is detected, this command automatically provides verbose information.

hwMonitor

Displays current port states, fan speeds, temperatures, and voltages.

Syntax:

```
hwMonitor slot
```

Options:

```
[slot]
```

Slot number to display. 0 for current slot of the management board.

Sample Output:

```
-> hwMonitor 0
[2J[0;0H    [7mSystem monitor, Uptime: 1 days 23 hours, 33 minutes[27m
  PS1  Fan Speed: [5m   -1 RPM[0m PS2  Fan Speed: [5m   -1 RPM[0m PS3  Fan Speed:
[5m   -1 RPM[0m
  PS4  Fan Speed:  9642 RPM[0m PS5  Fan Speed:  9642 RPM[0m PS6  Fan Speed:  9507
RPM[0m
  FT1 Fan Speeds:  8940 RPM[0m  8820 RPM[0m FT2 Fan Speeds:  9120 RPM[0m  8880
RPM[0m
  FT3 Fan Speeds:  9060 RPM[0m  8700 RPM[0m FT4 Fan Speeds:  8640 RPM[0m  8760
RPM[0m
  Device          T1    T2    A1    A2    1.2V     1.6V     1.8V
     Spine 1      34C   38C   26C   30C   1.226    1.489    1.773
     Spine 2      41C   39C   28C   28C   1.240    1.499    1.764
     Spine 3      36C   41C   24C   27C   1.226    1.509    1.770
     Leaf 1       ---   ---   ---   ---   -----    -----    -----
     Leaf 2       ---   ---   ---   ---   -----    -----    -----
     Leaf 3       ---   ---   ---   ---   -----    -----    -----
     Leaf 4       ---   ---   ---   ---   -----    -----    -----
     Leaf 5       ---   ---   ---   ---   -----    -----    -----
     Leaf 6       ---   ---   ---   ---   -----    -----    -----
     Leaf 8       ---   ---   ---   ---   -----    -----    -----
     Leaf 9       ---   ---   ---   ---   -----    -----    -----
     Leaf 11      38C   33C   26C   27C   1.184    1.499    1.783
     Leaf 12      ---   ---   ---   ---   -----    -----    -----
```

NOTES: To exit monitoring, press the enter key.

showIBNodeDesc

Displays the IB Node subnet management agent (SMA) Description..

Syntax:

```
showIBNodeDesc [-d]
```

Options:

[-d]

Show the default Node Name for this unit.

Sample Output:

```
-> showIBNodeDesc
IB Node (SMA) Description is = SilverStorm 9240 GUID=0x00066a000300012a
```

NOTES: Entering this command with no parameters displays the current Node Description. The -d option will display the unit's default value.

setIBNodeDesc

Changes the IB Node (SMA) Description.

Syntax:

```
setIBNodeDesc "New Node String"
```

Options:

None.

Sample Output:

```
-> setIBNodeDesc "SilverStorm 9080 GUID=0x00066a00da000"
IB Node (SMA) Description successfully changed to = SilverStorm 9080
```

NOTES: New node name must be enclosed in quotes. Node name must be less than 128 characters.

setIBNodeDescFormat

Changes the IB Node (SMA) description format modifier.

Syntax:

```
setIBNodeDescFormat [format]
```

Options:

None.

Sample Output:

```
-> setIBNodeDescFormat
Format = 0
```

NOTES: If the format is not specified, then the current format selection value is displayed. When the format is 0, the SMA node description is in a verbose format. When the format is one the format is brief. The brief format is more consistent with the CLI/GUI Port Stat port naming.

fruInfo

Displays the EEProm contents.

Syntax:

```
fruInfo [slot|'-all']
```

Options:

[slot]

Slot that you wish to display EEProm contents. Defaults to 0 (chassis EEProm)

NOTE:The '-all' parameter performs this command against all available slots for this device.

Sample Output:

```
-> fruInfo

Display chassis info
xInfo_ChassisInfo:
RecType:       1LastRec:       0LenMult:       0
ReadOnly:      1RecordFormat:  2RecLen:       14
LogicalLen:  020
HdrChkSum:    c6
ChassisGuid:  00066a000300012a
SlotCount:     1(IB Mods in Chassis)
SlotNumbers:  81(pairs:ext0|Slt1)
CmeAccess:    80(bits:ext0|Slt1|Cme)
SlotNumber:    0 <-Record accessed via this slot
CmeAccessBits: 2(Access slot relative)
ProxyAccess:   0(Access slot relative)
LockDrivesCTR: 0(Clear to Remove interlock)
MechLock:      1
NodeCount:     0
```

NOTES: The '-all' parameter performs this command against all available slots for the device.

chassisQuery

Displays information about modules inserted in the chassis/hemisphere that support firmware updates. This includes information regarding cardType, Board Support Package (BSP) and SlotNumber.

Syntax:

```
chassisQuery [slot] [-showType] [-type card_type]
[-ignoreInvalidType]
```

Options:

[slot]

slot number

[-showType]

display the card type

[-type card_type]

display slots that have given card type

[-ignoreInvalidType]

do not return an error if an invalid card type is supplied

NOTE:Entering this command with no parameters will display all currently occupied card slots.

Sample Output:

```
-> chassisQuery
slots: 2 4 8 101 102

-> chassisQuery 101 -showType
type: InfinIO9000.t3
```

NOTES: Entering this command with no parameters will display all currently occupied card slots.

showInventory

Displays a detailed list of all hardware within the chassis.

Syntax:

```
showInventory
```

Options:

None.

Sample Output:

```
-> showInventory


_____________________________________
             Leaf 7
-------------------------------------
        SlotNumber - 7
              GUID - 00066A00070001B6
   Manufacturer Id - 00066a
 Manufacturer Name - SST
     Mfg Date/Time - 2006/04/02  10:00
             Model - 9xxx-Leaf-4x-DDR
       Part Number - 220030-100-1
      Product Name - 9xxx DDR Leaf
     Serial Number - CNV1140600248
           Version - 100-1

...
```

Network

ifShow

Displays information for all the network interfaces for the switch, including the management port and the internal Ethernet network.

Syntax:

```
ifShow [ifName]
```

Options:

[ifName]

The network interface name.

Sample Output:

```
-> ifShow lo2 (unit number 0):
        Flags: (0x8069) UP LOOPBACK MULTICAST ARP RUNNING
        Type: SOFTWARE_LOOPBACK
        Internet address: 127.0.0.1
        Netmask 0xff000000 Subnetmask 0xff000000
        Net 0x7f000000 Subnet 0x7f000000
        Metric is 0
        Maximum Transfer Unit size is 32768
dp0 (mgmt_eth1 is active), (mgmt_eth2 is disabled):
        Flags: (0x8863) UP BROADCAST MULTICAST ARP RUNNING
        Type: ETHERNET_CSMACD
        Internet address: 172.26.0.221
        Broadcast address: 172.26.15.255
        Netmask 0xffff0000 Subnetmask 0xfffff000
        Net 0xac1a0000 Subnet 0xac1a0000
        Ethernet address is 00:06:6a:00:50:ba
        Metric is 0
        Maximum Transfer Unit size is 1500
        9081806 octets received
        1425622 octets sent
        153208 packets received
        18129 packets sent
        150530 broadcast packets received
        88 broadcast packets sent
        0 multicast packets received
        0 multicast packets sent
        0 input discards
        0 input unknown protocols
        0 input errors
        0 output errors
```

NOTES: This routine displays the attached network interfaces for debugging and diagnostic purposes. If ifName is given, only the interfaces belonging to that group are displayed. If ifName is omitted, all attached interfaces are displayed.

routeShow

Displays the OOB LAN IP routes.

Syntax:

```
routeShow
```

Options:

None.

Sample Output:

```
-> routeShow

ROUTE NET TABLE
destination      gateway              flags  Refcnt  Use
Interface
------------------------------------------------------------------------
0.0.0.0          172.26.0.254         3      2       1470
mgmt_eth1
172.26.0.0       172.26.0.221         101    0       0
mgmt_eth1
------------------------------------------------------------------------

ROUTE HOST TABLE
destination      gateway              flags  Refcnt  Use
Interface
------------------------------------------------------------------------
127.0.0.1        127.0.0.1            5      2       593           lo0
------------------------------------------------------------------------
```

ping

Send ping packets to a specified host.

Syntax:

```
ping [ipAddress] [packetCount]
```

Options:

[IpAddress]

The IP address of the network host to ping.

[packetCount]

The number of packets with which to ping the host (default is 5).

Sample Output:

```
PING 172.26.0.254: 56 data bytes
64 bytes from 172.26.0.254: icmp_seq=0. time=0. ms
64 bytes from 172.26.0.254: icmp_seq=1. time=0. ms
64 bytes from 172.26.0.254: icmp_seq=2. time=0. ms
64 bytes from 172.26.0.254: icmp_seq=3. time=0. ms
64 bytes from 172.26.0.254: icmp_seq=4. time=0. ms
----172.26.0.254 PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 0/0/0
```

NOTES: This routine spawns a process to send ping packets to the specified IP address. If packetCount is given, the process exits after that number of packets are sent. If packetCount is ommitted, a default the number of packets defaults to 5.

showChassisIpAddr

Displays the chassis IP address.

Syntax:

```
showChassisIpAddr
```

Options:

None.

Sample Output:

```
-> showChassisIpAddr
Chassis IP Address: 172.26.0.221 Net mask: 255.255.240.0
```

setChassisIpAddr

Change the OOB LAN IP address and network mask.

Syntax:

```
setChassisIpAddr -h ipaddress -m netMask
```

Options:

-h ipaddress

The new IP address in dotted notation format 'xxx.xxx.xxx.xxx'.

-m netMask

The network mask. (may be in dotted notation or hexidecimal format)

Sample Output:

```
-> setChassisIpAddr -h 172.26.0.221 -m 255.255.240.0
```

NOTES: Changing the chassis IP address will drop the connection to the CLI if not using the console port, and may cause the device to become unreachable.

showDefaultRoute

Displays the default gateway IP address.

Syntax:

```
showDefaultRoute
```

Options:

None.

Sample Output:

```
-> showDefaultRoute
Gateway IP Address: 172.26.0.254
```

NOTES: This is the IP address for the default gateway to route packets from the OOB mangement port to an external network.

setDefaultRoute

Change the default gateway IP address.

Syntax:

```
setDefaultRoute -h ipaddress
```

Options:

-h ipaddress

The default gateway IP address in dotted decimal format (xxx.xxx.xxx.xxx).

Sample Output:

```
setDefaultRoute -h 172.26.0.235
```

NOTES: This allows the user to configure the IP address for the default gateway to route packets from the OOB management port to an external network.

arpShow

Displays the contents of the ARP table.

Syntax:

```
arpShow
```

Options:

None.

Sample Output:

```
-> arpShow

LINK LEVEL ARP TABLE
destination      gateway            flags  Refcnt  Use           Interface
------------------------------------------------------------------------------
172.26.0.203     00:11:25:c3:07:c5    405    0       0             mgmt_eth1
172.26.0.254     00:11:25:c3:07:c5    405    1       1419          mgmt_eth1
------------------------------------------------------------------------------
```

hostShow

Displays a list of remote hosts names, internet address and alias(es).

Syntax:

```
hostShow
```

Options:

None.

Sample Output:

```
-> hostShow
hostname          inet address       aliases
--------          ------------       -------
localhost         127.0.0.1
home              10.90.90.0
slot1             127.1.0.1
slot2             127.1.0.2
slot3             127.1.0.3
slot4             127.1.0.4
slot5             127.1.0.5
slot6             127.1.0.6
slot7             127.1.0.7
slot8             127.1.0.8
slot9             127.1.0.9
slot10            127.1.0.10
slot11            127.1.0.11
slot12            127.1.0.12
slot13            127.1.0.13
slot14            127.1.0.14
slot15            127.1.0.15
slot16            127.1.0.16
slot17            127.1.0.17
slot18            127.1.0.18
slot19            127.1.0.19
slot20            127.1.0.20
slot21            127.1.0.21
slot22            127.1.0.22
slot23            127.1.0.23
slot24            127.1.0.24
switchA           127.1.1.1          spine1
switchB           127.1.1.2          spine2
switchC           127.1.1.3          spine3
Master            127.2.0.101
Slave             127.2.0.102
```

Firmware

fwUpdate (9024 Only)

Updates the firmware.

Syntax:

```
fwUpdate [host user password dir filename]
```

Options:

host
The name of the host where the firmware file resides.

user
The FTP user name.

password
The FTP user password.

dir
After logging in, the directory to **cd** to.

filename
The name of the firmware file.

Sample Output:

```
-> fwupdate
Enter 1 for FTP, 2 for local file: 1
Ftp Server IP Address:[0.0.0.0] 192.168.0.10
Ftp username:[ftp] ftp
Ftp password:[ftp] ftp
File Directory:[] /firmware/switches/
File name:[] InfinIO9000.t3.pkg
Save changes? [Y] n
Attempting to initiate firmware update
  Product               = InfinIO9000
  Version               = 4.2.1.1.1
  Compressed Image Size = 4837085 bytes
  md5                   = 9de07c302bb09c4335b114dc809b87ff
  vxWorks Image Type    = loadable
  Computed md5          = 9de07c302bb09c4335b114dc809b87ff
  md5 values match!
Firmware update initiated successfully
Info, update of all IDB+ boot image data completed successfully.
Image        = /image2
Product      = InfinIO9000
BSP          = t3
Version      = 4.2.1.1.1
Image Size   = 4837085 bytes
md5          = 9de07c302bb09c4335b114dc809b87ff
Computed md5 = 9de07c302bb09c4335b114dc809b87ff information
md5 values agree.
Info, update of all IDB+ boot image data completed successfully.

Firmware update completed successfully
```

NOTES: Firmware update works by using ftp to retrieve the firmware file, then writes the file to flash. Omitting any of the options causes the user to be prompted for the information.

fwUpdateSlot

Updates the firmware on a specific line card.

Syntax:

```
fwUpdateSlot slotnumber [host user password dir
filename]
```

Options:

slotnumber

The chassis slot numberto update.

host

The name of the host where the firmware file resides.

user

The FTP user name.

password

The FTP user password.

dir

After logging in, the directory to **cd** to.

filename

The name of the firmware file.

Sample Output:

```
-> fwUpdateSlot 1
Enter 1 for FTP, 2 for local file: 1
Ftp Server IP Address:[192.168.0.195]
Ftp username:[ftp] xxxxx
Ftp password:[ftp] xxxxx
File Directory:[PATH TO FIRMWARE FILE]
File name:[9000.pkg]
```

NOTES: Firmware update works by using ftp to retrieve the firmware file, then writes the file to flash. Omitting any of the options causes the user to be prompted for the information.

fwUpdateChassis

Updates the firmware for all cards in a chassis of a particular type, or everything.

Syntax:

```
fwUpdateChassis (all,management,evic, fvic) [noprompt]
[reboot]
```

Options:

(all,management,evic, fvic)

Type of card to update. See notes for details.

noprompt

Will not prompt the user for ftp information (uses the saved values).

reboot

Upon successful completion, reboots the updated cards .

Sample Output:

```
-> fwUpdateChassis all reboot
```

NOTES: Firmware update works by retrieving the firmware file from an ftp server. The file is saved locally and then written to flash. Using the 'all' option, all cards in the chassis will be updated. This assumes that all firmware files are in the same location. The user will be prompted for the names of each file for each line card type present. The system can save the values entered as defaults for future firmware updates.

fwListFiles

Lists the contents of the firmware directory.

Syntax:

```
fwListFiles
```

Options:

None.

Sample Output:

```
-> fwListFiles

Listing Directory /firmware:
[PATH TO FIRMWARE FILE] / [FIRMWARE FILE NAME]
```

NOTES: The firmware directory temporarily stores firmware files before they are written to flash.

fwShowUpdateParams

Display the default update firmware settings.

Syntax:

```
fwShowUpdateParams
```

Options:

None.

fwSetUpdateParams

Change the default update firmware settings.

Syntax:

```
fwSetUpdateParams -c cardtype [-h hostname] [-u
username] [-p password] [-d directory] [-f filename]
```

Options:

-c - cardtype

Choices are: 'manangement', 'vfx', 'vex', 'ibx'.

-h - hostname

The host name or IP address of the FTP server.

-u - username

The name of the user accessing the ftp server.

-p - password

The password of the user accessing the ftp server.

-d - directory

The directory containing the firmware file.

-f - filename

The firmware file name.

NOTES: Modifies the default update firmware parameters. Except for the filename option, all parameters are shared across all card types. The cardtype parameter is only required when specifying the filename.

showCapability

Display capability and feature information for a specific release.

Syntax:

```
showCapability [-key feature]
```

Options:

-key feature

Displays information for a particular feature.

Sample Output:

```
-> showCapability
fwPush: 1
```

showLastScpRetCode

Display the return code from the last SCP Firmware Push.

Syntax:

```
showLastScpRetCode slot | -all
```

Options:

slot

The slot number in the chassis.

-all

All slots in the chassis.

Sample Output:

```
-> showLastScpRetCode 101
SCP: Slot 101 Last Exit Code: 0: Success
```

NOTES: This allows for automated systems to determine if a SCP firmware push was successful or not.

fwVersion

Displays the firmware versions for a unit.

Syntax:

```
fwVersion [slot]
```

Options:

slot

Slot number.

Sample Output:

```
-> fwVersion 3
Slot   3 Information -----------------
  Firmware Version: 4.0.0.0.32
  Firmware build:   4_0_0_0_32
  Firmware BSP:     t3=9000 chassis
  MBC Version:      None
  Bootrom Version:  4.0.0.0.28
```

bootQuery

Displays boot image version information.

Syntax:

```
bootQuery slot [-active | -alternate | -all]
```

Options:

slot
Slot number.

-active
Displays the version of the active firmware image.

-alternate
Displays the version of the alternate firmware image.

-all
Displays the versions for the primary and alternate firmware images.

Sample Output:

```
-> bootQuery 2 -all
Primary firmware version: 4.2.2.0.2
Alternate firmware version: 4.2.1.0.2
Active firmware version: 4.2.2.0.2
```

bootSelect

Select the next boot image to be used.

Syntax:

```
bootSelect slot [-i index] [-alternate] [-version
version] [-noprompt]
```

Options:

slot
The slot number using the next boot image.

-i index
The index of the boot image to be used next.

-alternate
Chooses the alternate image to be used next.

-version version
Chooses a specific version to be the image to be used next.

-noprompt
Displays the current configuration only.

Sample Output:

```
-> bootSelect 3 -noprompt
Currently installed firmware versions
index : alias    : version
-------------------------
   1   : image1   : 4.0.0.0.28;
*# 2   : image2   : 4.0.0.0.32;

  * - indicates Default image (will run at next reboot)
  # - indicates Active image

Default boot image index = 2
```

NOTES: This command allows the user to set the next boot image for the device. A '*' next to the image entry indicates the currently selected bootimage. A '#' indicates the currently active bootimage.

Subnet Management

smControl

Starts and stops the embedded subnet manager.

Syntax:

```
smControl start | stop | restart | status
```

Options:

Start

Start the SM.

Stop

Stop the SM.

restart

Restarts the SM.

status

Prints out the SM Status.

Sample Output:

```
-> smControl start
Starting the SM...
```

smConfig

Configure startup parameters of the embedded subnet manager.

Syntax:

```
smConfig [startAtBoot yes|no] [startOnSlaveCmu yes|no]
```

Options:

startAtBoot

Start the subnet manager at chassis boot

startOnSlaveCmu

Start subnet manager on the slave CMU

Sample Output 1:

```
-> smConfig
Start at boot? [Y]
Start on slave CMU? [N]
MasterSpine1->
```

Sample Output 2:

```
-> smconfig startAtBoot yes startOnSlaveCmu yes
Saving....
Saving complete...
```

NOTES: Use this command to configure the subnet manager. Note that for runtimetype parameters the Subnet manager may need to be restarted for them to take effect.

smShowLids

Display all fabric LID information as known by the subnet manager.

Syntax:

```
smShowLids
```

Options:

None.

Sample Output:

```
sm_state = MASTER   count = 572781   LMC = 0, Topology Pass count = 339, Priority = 0, Mkey = 0x0

-------------------------------------------------------------------------------------------
SilverStorm 9080 GUID=0x00066a00da000100 172.26.2.2 Spine 1, Ch
-------------------------------------------------------------------------------------------
Node[  0] => 00066a000600013c  (2)  ports=24, path=
       Port ----  GUID  ----  (S)   LID       LMC     _VL_   __MTU__  _WIDTH_  ____SPEED____  CAP_MASK   N#  P#
          0 00066a000600013c   4  LID=0001  LMC=0000   8  8    2k  2k   4X  4X   2.5     2.5   00000a4a    0   0
          4 0000000000000000   4                        8  8    2k  2k   4X  4X  2.5/5    5.0   00000000    1
22  4 22
          5 0000000000000000   4                        8  8    2k  2k   4X  4X  2.5/5    5.0   00000000    1
23  4 23
          6 0000000000000000   4                        8  8    2k  2k   4X  4X  2.5/5    5.0   00000000    1
24  4 24
          7 0000000000000000   4                        8  8    2k  2k   4X  4X  2.5/5    5.0   00000000    2
24  7 24
          8 0000000000000000   4                        8  8    2k  2k   4X  4X  2.5/5    5.0   00000000    2
23  7 23
          9 0000000000000000   4                        8  8    2k  2k   4X  4X  2.5/5    5.0   00000000    2
22  7 22
         22 0000000000000000   4                        8  8    2k  2k   4X  4X  2.5/5    5.0   00000000    3
16 22 16
         23 0000000000000000   4                        8  8    2k  2k   4X  4X  2.5/5    5.0   00000000    3
18 22 18
         24 0000000000000000   4                        8  8    2k  2k   4X  4X  2.5/5    5.0   00000000    3
17 22 17

-------------------------------------------------------------------------------------------
FVIC in Chassis 0x00066a00da000100, Slot 7
-------------------------------------------------------------------------------------------
Node[  4] => 00066a10dd00004a  (1)  ports=1, path= 4 11
       Port ----  GUID  ----  (S)   LID       LMC     _VL_   __MTU__  _WIDTH_  ____SPEED____  CAP_MASK   N#  P#
          1 00066a11dd00004a   4  LID=0004  LMC=0000   4  4    2k  2k   4X  4X  2.5/5    5.0   02090048    1
11  4 11

-------------------------------------------------------------------------------------------
st19
-------------------------------------------------------------------------------------------
Node[ 10] => 00066a009800035a  (1)  ports=2, path= 7  4
       Port ----  GUID  ----  (S)   LID       LMC     _VL_   __MTU__  _WIDTH_  ____SPEED____  CAP_MASK   N#  P#
          1 00066a00a000035a   4  LID=009f  LMC=0000   4  4    2k  2k   4X  4X   2.5     2.5   02010048    2
4  7  4
```

NOTES: Use this command to display the current LID assignments for the devices in the InfiniBand fabric.

smPriority

Set/display the priority of the subnet manager.

Syntax:

```
smPriority
```

Options:

priority-value
priority must be between 0 and 15, inclusive.

Sample Output 1:

```
-> smPriority
The SM Priority is 0
```

Sample Output 2:

```
-> smPriority 4
SM Priority has been set to 4
```

NOTES: The priority of the Subnet Manager(SM) determines which subnet manager will become the master SM for the fabric. Zero is the lowest priority and fifteen is the highest. The SM with the highest priority will become the master SM for the fabric.

smSweepRate

Set/display the sweep rate of the subnet manager.

Syntax:

```
smSweepRate [sweepRate]
```

Options:

sweepRate
The sweep rate (in seconds) of the subnet manager. Valid values are 3-86400, or 0 to turn the sweep off. The sweepRate is the interval between the end of one sweep and the start of the next sweep.

Sample Output:

```
-> smSweepRate 300
The SM sweep rate has been set to 300 seconds
```

NOTES: The sweep rate determines how often the subnet manager scans the fabric for changes and events.

bmSetSweepRate

Set/display the sweep rate of the baseboard manager.

Syntax:

```
bmSetSweepRate [sweepRate}
```

Options:

sweepRate

The sweep rate (in seconds) of the baseboard manager. Valid values are 30-86400, or 0 to turn the sweep off.

Sample Output 1:

```
-> bmsweeprate
The BM sweep rate is 86400 seconds
```

Sample Output 2:

```
-> bmsweeprate 500
The BM sweep rate has been set to 500 seconds
```

NOTES: The sweep rate determines how often the baseboard manager scans the fabric for changes and events.

smForceSweep

Force a fabric sweep by the embedded subnet manager.

Syntax:

```
smForceSweep
```

Options:

None.

Sample Output:

```
-> smForceSweep
```

NOTES: This command has no output message. To see the resulting sweep information, the "Info" level log messages must be turned on.

bmForceSweep

Force a fabric sweep by the embedded baseboard manager.

Syntax:

```
bmForceSweep
```

Options:

None.

Sample Output:

```
-> bmForceSweep
```

NOTES: Use this command to force a sweep by the baseboard manager.

smShowGroups

Display multicast group information in the embedded subnet manager.

Syntax:

```
smShowGroups [-h]
```

Options:

-h
Display the host name as part of the output.

Sample Output:

```
-> smShowGroups
Multicast Groups:
  join state key: F=Full N=Non S=SendOnly Member

0xff12601bffff0000:00000001ffffd5bb (c001)
  qKey = 0x00000000  pKey = 0xFFFF  mtu = 4  rate = 3  life = 19  sl = 0
  0x0011750000ffd5bb F

0xff12401bffff0000:00000000ffffffff (c000)
  qKey = 0x00000000  pKey = 0xFFFF  mtu = 4  rate = 3  life = 19  sl = 0
  0x00066a01a0007116 F  0x0002c902003fffd5 F  0x00066a00a00001ac F
  0x00066a01a000015d F  0x00066a00a00001a3 F  0x00066a00a00001dc F
  0x00066a00a000035a F  0x0011750000ffd5c2 F  0x0011750000ffd664 F
  0x0011750000ffd9c2 F  0x0011750000ffd9f8 F  0x0011750000ffd5b9 F
  0x0011750000ffda4a F  0x0011750000ffd5bb F  0x0011750000ffd9de F
```

NOTES: Use this command to display multicast group information in the subnet manager.

smShowServices

Display subnet administration service records of the subnet manager.

Syntax:

```
smShowServices
```

Options:

None.

Sample Output:

```
-> smShowServices
*****************************************************************
                  There is 1 Service Records
*****************************************************
Service ID          = 0x1100D03C34834444
Service GID         = 0xFE80000000000000:00066A000600013C
Service P_Key       = 0x0000
Service Lease       = infinite
Service Key         =
0x00 0x00 0x00 0x00  0x00 0x00 0x00 0x00  0x00 0x00 0x00 0x00  0x00
0x00 0x00 0x00
Service Name        = SilverStorm Fabric Executive service Rev 1.1
Service Data 8      =
0x00 0x00 0x00 0x00  0x00 0x00 0x00 0x00  0x00 0x00 0x00 0x00  0x00
0x00 0x00 0x00
Service Data 16     =
0x0000 0x0000  0x0000 0x0000  0x0000 0x0000  0x0000 0x0000
Service Data 32     =
0x0000 0x0000 0x0000 0x0000
Service Data 64     =
0x0000000000000000 0x0000000000000000
Service Expire Time = 0x0100000000000000
```

NOTES: The components(fields) of each service record are displayed. Each service record is stored in a location identified by a 'Slot' number which is displayed before any component of that Service Record. If a group of slots do not contain Service Records, the first slot of the empty group is displayed as 'empty'.

smShowSubscriptions

Display event forwarding (subscription) table in the embedded subnet manager.

Syntax:

```
smShowSubscriptions
```

Options:

None.

Sample Output:

```
-> smShowSubscriptions
*****************************************************************
                There are 2 subscriptions
*****************************************************************
Subscriber GID         = 0xFE80000000000000:00066A00D8000163
Subscriber LID         = 0x0071
Subscriber PKey        = 0xFFFF
Subscriber Start LID   = 0x0001
Subscriber End LID     = 0xBFFF
Subscriber Record ID   = 0x00000001
Subscriber Inform Info =
GID                  = 0x0000000000000000:0000000000000000
Start LID            = 0xFFFF
End LID              = 0x0000
Is Generic?          = Yes
Subscribe?           = Subscribe
Type                 = All Types
Trap Number            = 0x0040
Queue Pair Number      = 0x000001
Response Time Value    = 19
Producer Type          = Subnet Management
*****************************************************************
Subscriber GID         = 0xFE80000000000000:00066A01A0007116
Subscriber LID         = 0x0007
Subscriber PKey        = 0xFFFF
Subscriber Start LID   = 0x0001
Subscriber End LID     = 0xBFFF
Subscriber Record ID   = 0x00000036
Subscriber Inform Info =
GID                  = 0x0000000000000000:0000000000000000
Start LID            = 0xFFFF
End LID              = 0x0000
Is Generic?          = Yes
Subscribe?           = Subscribe
Type                 = All Types
Trap Number            = 0x0043
Queue Pair Number      = 0x000001
Response Time Value    = 18
Producer Type          = Channel Adapter
*****************************************************************
                There are 2 subscriptions
```

NOTES: Use this command to display the event forwarding (subscription) table in the subnet manager.

smMasterLMC

Set/display the Master SM's LMC value to be used on CA ports.

Syntax:

```
smMasterLMC [lmc]
```

Options:

lmc
The LMC value between 0 and 7, inclusive.

Sample Output:

```
-> smMasterLMC 2
Master SM LMC: 2 (4 LID(s) per port)
```

NOTES: The value of the LMC determines how many LID's are assigned to an endport; 2 LMC LIDs are assigned to endports based on this value. For example, setting the LMC to a value of 3 will assign 2^3 or 8 LID's per endport.

smShowMasterLid

Display the LID of the subnet manager.

Syntax:

```
smShowMasterLid
```

Options:

None.

Sample Output:

```
-> smShowMasterLid
The SM LID is 0x0001
```

NOTE: Use this command to display the SM's LID. It may be the local LID if the SM is the master.

smShowLidMap

Display the LID-to-port GUID map for the subnet manager.

Syntax:

```
smShowLidMap
```

Options:

None.

Sample Output:

```
-> smShowLidMap
------------------------------------------------------------------------
------------
SM is currently in the MASTER state, with Topology Pass count = 341
------------------------------------------------------------------------
------------
Lid 0x0001: guid = 0x00066a000600013c, pass = 341, SilverStorm 9080
GUID=0x00066a00da000100 172.26.2.2 Spine 1, Ch
Lid 0x0002: guid = 0x00066a0007000170, pass = 341, SilverStorm 9080
GUID=0x00066a00da000100 172.26.2.2 Leaf 4, Chi
Lid 0x0003: guid = 0x00066a100600013c, pass = 341, SilverStorm 9080
GUID=0x00066a00da000100 172.26.2.2 Spine 1, Ch
Lid 0x0004: guid = 0x00066a11dd00004a, pass = 341, FVIC in Chassis
0x00066a00da000100, Slot 7
Lid 0x0005: guid = 0x00066a21dd00004a, pass = 341, FVIC in Chassis
0x00066a00da000100, Slot 7
Lid 0x0006: guid = 0x00066a00a0000248, pass = 229
Lid 0x0007: guid = 0x00066a01a0007116, pass = 341, st149
Lid 0x0008: guid = 0x0000000000000000, pass = 0
Lid 0x0027: guid = 0x00066a026000016c, pass = 341, VFx in Chassis
0x00066a0050000135, Slot 5
Lid 0x0028: guid = 0x0000000000000000, pass = 0
Lid 0x0029: guid = 0x00066a0260000174, pass = 341, VFx in Chassis
0x00066a000100024d, Slot 2
Lid 0x002a: guid = 0x0000000000000000, pass = 0
```

NOTES: Use this command to display the LID-to-port GUID map of the subnet manager. The pass count for a LID is incremented each time the SM sweep detects that LID.

If LMC has been used to assign multiple LIDs to a node, those assignments will be reflected in the smShowLidMap output.

smShowMaxLid

Display the highest LID allocated by the subnet manager.

Syntax:

```
smShowMaxLid
```

Options:

None.

Sample Output:

```
-> smShowMaxLid
The maximum LID is 0x0138
```

NOTES: Use this command to display the highest LID allocated by the subnet manager.

smSwitchLifetime

Set/Display the default switch lifetime in the SM.

Syntax:

```
smSwitchLifetime [lifetime}
```

Options:

lifetime

The packet lifetime value between 0 and 31, inclusive.

Sample Output:

```
-> smSwitchLifetime 15
SM switch packet lifetime: 15 (~134217 microseconds)
```

NOTES: The switch lifetime value determines the maximum time a packet may remain in a switch, calculated using the formula: 4.096 * (2 ^ switchlifetime) microseconds.

smHoqLife

Set/Display the head of queue packet lifetime for switch ports.

Syntax:

```
smHoqLife [lifetime]
```

Options:

lifetime
The packet lifetime value between 0 and 31, inclusive.

Sample Output:

```
-> smHoqLife 9
SM HOQ Liftime: 9 (~2097 microseconds)
```

NOTES: Use this command to set the maximum lifetime that a packet may remain at the head of virtual lane's transmission queue before it is discarded by a switch, calculated using the formula: 4.096 * (2 ^ switchlifetime) microseconds.

smVLStall

Set/Display the VL stall value in the SM.

Syntax:

```
smVLStall [packets]
```

Options:

packets
The number of sequential packets dropped before port enters VL stalled state.

Sample Output:

```
-> smVLStall 5
SM VL Stall Threshhold: 5 packets
```

NOTES: Use this command to set the VL stall value for ports in the fabric. This value determines the how quickly a virtual lane for a particular switch or endport enters a 'stalled' state after dropping packets.

smShowSMParms

Display subnet manager parameters switch lifetime, HOQ lifetime, VLStall val, pkt lifetime, and dynamic PLT.

Syntax:

```
smShowSMParms
```

Options:

None.

Sample Output:

```
-> smShowSMParms
SM priority is set to 4
SM LMC is set to 0
SM sweep rate is set to 300
SM max retries on receive set to 3
SM max receive wait interval set to 250 millisecs
switchLifetime set to 15
HoqLife set to 9
VL Stall set to 5
packetLifetime constant is set to 18
Dynamic PLT ON using values: 1 hop=16, 2 hops=17, 3 hops=17, 4 hops=18,
5 hops=18, 6 hops=18, 7 hops=18, 8+hops=19
SM DBSync interval set to 900
SM topology errors threshold set to 0, max retry to 3
```

NOTES: Use this command to display a sampling of subnet manager parameters.

smPKeys

Configure a partition key (PKey) in the PKey table.

Syntax:

```
smPKeys [index] [pkey] [description]
```

Options:

index

PKey index.

pkey

Pkey 16 Bit value.

description

User-defined description of the PKey.

Sample Output:

```
-> smPKeys 0 0xffff "Default PKey"
Successfully set PKey index: 0 to 0xffff
```

NOTES: PKeys are used for partitioning the subnet. Only configure PKeys if the host driver supports this. Invalid configuration of the PKey may render the fabric inoperable.

smInfoKey

Set the subnet manager key (SMInfo) value.

Syntax:

```
smInfoKey [key]
```

Options:

key
The SmInfo key (8 byte value in hex or decimal).

Sample Output:

```
-> smInfoKey 0x1
SM Key: 0x0000000000000001 (1 decimal).
```

NOTES: Use this command to set the SM key. SM must be offline and key value is up to 8 byte hex.

smMgmtKey

Set the subnet manager management key (portInfo) value.

Syntax:

```
smMgmtKey [mKey]
```

Options:

mKey
The management key (8 byte value in hex or decimal)

Sample Output:

```
-> smMgmtKey 0x11
SM management key: 0x0000000000000011 (17 decimal)
```

NOTES: Use this command to set the SM management key. SM must be offline and mkey value is up to 8 byte hex.

smOptionConfig

Use this command to configure support for non-default modes of operation.

Syntax:

```
smOptionConfig [clear | default | [def-mcgrp-create]
[dyn-plt]]
```

Options:

clear

Clears all set options.

default

Enable the default set of options (i.e., def-mcgrp-create and dyn-pit).

def-mcgrp-create

Auto create default multicast group. If using only this option, dyn-pit will be disabled.

dyn-pit

Enable dynamic packet lifetimes for pathrecord queries. If using only this option, def-mcgrp-create will be disabled.

Sample Output:

```
-> smOptionConfig dyn-plt
        [dyn-plt] Dynamic packet lifetime support is enabled
```

NOTES: Use of these options without explicit direction may cause the fabric to be inoperable.

smDefBcGroup

Set/display default multicast group configuration.

Syntax:

```
smDefBcGroup [enable | disable] [default | pKey [mtu
[rate [sl [qKey] [fl [tc]
```

Options:

enable
Enable auto-creation of multicast group at SM startup.

disable
Disable auto-creation of multicast group at SM startup.

default
Set PKey, MTU, rate, SL, QKey FlowLabel, and TClass to default values.

pkey
Partition Key. If no value entered, defaults to use the default PKey (0xFFFF).

mtu
Maximum transfer unit. If no value entered, defaults to use mtu of 2048.

rate
Data Rate. If no value is entered, defaults to use a rate of 10GB.

sl
Service level. If no value is entered, defaults to service level 0.

qKey
Queue Key. If no value is entered, defaults to 0.

FlowLabel
Flow Label. If no value is entered, defaults to 0.

TClass
Traffic Class. If no value entered, defaults to 0.

Sample Output:

```
-> smDefBcGroup 0xffff 4 3 0
```

NOTES: This will enable or disable auto-creation of a default broadcast group with one member with a GUID of 0x00066A00FACADE01. Executing this command with with the argument 'default' sets the default broadcast group with pkey=0xffff, MTU=4 (2048), RATE=3 (10GB), SL=0, QKey=0, FlowLabel=0, TClass=0. If auto-creation was disabled and you choose to enable auto-creation while the SM is running, the group will be created immediately.

- Valid MTU values are 1(256), 2(512), 3(1024), 4(2048), and 5(4096)

- Valid RATE values are 2(2.5GB), 3(10GB), 4(30GB), 5(5GB), 6(20GB), 7(40GB), 8(60GB), 9(80GB), 10(120GB)
- Valid Values for SL are 0 to 0xF
- Valid Values for QKEY are 0 to 0xFFFFFFFF
- Valid Values for FlowLabel are 0 to 0xFFFFF
- Valid Values for TClass are 0 to 0xff

NOTE:DO NOT CREATE THIS GROUP WITH PARAMETERS THAT YOUR FABRIC CANNOT SUPPORT!

smGidPrefix

Set the Subnet Prefix (default=0xfe80000000000000).

Syntax:

```
smGidPrefix [prefix]
```

Options:

prefix
The prefix to use when assigning GIDs to nodes.

Sample Output:

```
-> smGidPrefix
Subnet Prefix: 0xfe80000000000001
```

NOTES: Use this command to set the subnet prefix of the SM. The SM must be offline and the subnet prefix value is 8 byte hex.

smSubnetSize

Set/display the subnet size for the subnet manager.

Syntax:

```
smSetSubnetSize subnetSize
```

Options:

subnetSize
The number of end ports on the subnet; must be between 40 and 288, inclusive.

Sample Output:

```
-> smSubnetSize 288
Subnet Size: 288
```

NOTES: Use this command to tune the SM to handle the configured fabric size. This should be expressed in terms of the upper limit of HCA ports on the subnet. Setting this value will not take effect until the Subnet Manager is restarted.

smTopoErrorThresh

Set/display the error threshold for a topology sweep.

Syntax:

```
smTopoErrorThresh [threshold]
```

Options:

None.

Sample Output:

```
-> smTopoErrorThresh 100
Set topology error threshold to:100
```

NOTES: Sets the maximum number of errors the SM may encounter during a sweep before abandoning the sweep.

smTopoAbandonThresh

Set/display the max consecutive times the SM can abandon a sweep due to too many errors.

Syntax:

```
smTopoAbandonThresh [threshold]
```

Options:

None.

Sample Output:

```
-> smTopoAbandonThresh 3
Set topology sweep abandonment threshold to:3
```

smMaxRetries

Set/display maximum number of SM receive retries.

Syntax:

```
smMaxRetries [retries]
```

Options:

None.

Sample Output:

```
-> smMaxRetries 3
Set max retries to:3

-> smMaxRetries
Max retries: 3
```

smRcvWaitTime

Set/display max time to wait for a reply to an SM packet in millisecs.

Syntax:

```
smRcvWaitTime [msecs]
```

Options:

None.

Sample Output:

```
-> smRcvWaitTime 250
Set recieve wait time to: 250 milliseconds

-> smRcvWaitTime
Recieve wait time: 250 milliseconds
```

smNonRespDropTime

Set/display seconds to wait before dropping a non-responsive node.

Syntax:

```
smNonRespDropTime [seconds]
```

Options:

None.

Sample Output:

```
-> smNonRespDropTime 300
Set Nonresponsive node drop time to: 300 seconds

-> smNonRespDropTime
Non-responsive node drop time: 300 seconds
```

smNonRespDropSweeps

Set/display sweeps to wait before dropping a non-responsive node.

Syntax:

```
smNonRespDropSweeps [sweeps]
```

Options:

None.

Sample Output:

```
-> smNonRespDropSweeps 3
Set Nonresponsive node drop sweeps to: 3 sweeps

-> smNonRespDropSweeps
Non-responsive node drop sweeps: 3 sweeps
```

smLogLevel (9000 MPFD Series Only)

Set and display log level settings.

Syntax:

```
smLogLevel [level]
```

Options:

None.

Sample Output:

```
-> smLogLevel 1
```

NOTES: Level settings are 1 for errors and warnings; 2 for errors, warnings and sweep info. Values 3, 4 and 5 are unused.

smMcLidTableCap

Set/display the limit of multicast LIDs available for allocation.

Syntax:

```
smMcLidTableCap [limit]
```

Options:

None.

Sample Output:

```
-> smMcLidTableCap 1024
Set mc lid limit to 1024

-> smMcLidTableCap
Mc lid limit: 1024
```

NOTES: Setting this value to zero disables limiting multicast LIDs.

smMasterPingInterval

Set/displays SM ping interval in seconds.

Syntax:

```
smMasterPingInterval [seconds]
```

Options:

None.

Sample Output:

```
-> smMasterPingInterval 4
Set master ping interval to 3 seconds

-> smMasterPingInterval
Master ping interval: 4 seconds
```

NOTES: Value must be between 3 and 10.

smMasterPingFailures

Set/display number of master ping failures allowed.

Syntax:

```
smMasterPingFailures [failures]
```

Options:

None.

Sample Output:

```
-> smMasterPingFailures 3
Set master ping failures to 3 failures

-> smMasterPingFailures
Master ping failures: 3 failures
```

NOTES: Value must be between 2 and 5.

smDbSyncInterval

Set/display how often a Master SM should perform a full sync with standby SMs.

Syntax:

```
smDbSyncInterval [minutes]
```

Options:

None.

Sample Output:

```
-> smDbSyncInterval 15
Set db sync interval to 15 minutes

-> smDbSyncInterval
SM DB full sync interval currently set to 15 minutes

-----SM DB SYNCHRONIZATION interval set to 900 seconds, 2 SM(s) in
fabric-----
MASTER SM node at SilverStorm 9024 DDR GUID=0x00066a00d90003fa, LID
0x0008, PortGuid 0x00066a00d90003fa
     Sync Capability is  SUPPORTED
STANDBY SM node at st44, LID 0x0100, PortGuid 0x00066a00a0000357
     Sync Capability is  SUPPORTED
     Full sync status is      SYNCHRONIZED
     Time of last Full sync is THU APR 10 15:37:47 2008
     Time of last INFORM records sync is THU APR 10 15:37:47 2008
     Time of last GROUP records sync is THU APR 10 15:37:47 2008
     Time of last SERVICE records sync is THU APR 10 15:37:47 2008
```

NOTES: Value must be between 0 and 60 minutes (0=OFF).

smDynamicPlt

Set/display dynamic packet lifetime values.

Syntax:

```
smDynamicPlt [index] [plt]
```

Options:

index

The index of the entry in the table to be changed. Range is 1-9..

plt

The packet lifetime value. The range is 0-63.

Sample Output:

```
-> smDynamicPlt
Index: 1 PLT Value: 16 (~268435 usec)
Index: 2 PLT Value: 17 (~536870 usec)
Index: 3 PLT Value: 17 (~536870 usec)
Index: 4 PLT Value: 18 (~1073741 usec)
Index: 5 PLT Value: 18 (~1073741 usec)
Index: 6 PLT Value: 18 (~1073741 usec)
Index: 7 PLT Value: 18 (~1073741 usec)
Index: 8 PLT Value: 19 (~2147483 usec)
Index: 9 PLT Value: 19 (~2147483 usec)

Dynamic packet lifetime values for pathrecord queries are enabled
(use the smOptionConfig command to change)



-> smDynamicPlt 1 15
Index: 1 PLT Value: 15 (~134217 usec)
Index: 2 PLT Value: 17 (~536870 usec)
Index: 3 PLT Value: 17 (~536870 usec)
Index: 4 PLT Value: 18 (~1073741 usec)
Index: 5 PLT Value: 18 (~1073741 usec)
Index: 6 PLT Value: 18 (~1073741 usec)
Index: 7 PLT Value: 18 (~1073741 usec)
Index: 8 PLT Value: 19 (~2147483 usec)
Index: 9 PLT Value: 19 (~2147483 usec)

Dynamic packet lifetime values for pathrecord queries are enabled
(use the smOptionConfig command to change)
```

NOTES: Setting values to numbers greater than 19 give an effectively-infinite packet lifetime.

sm1xLinkMode

Set/display how the SM handles links that come up at 1x.

Syntax:

```
sm1xLinkMode off|ignore
```

Options:

None.

Sample Output:

```
-> sm1xLinkMode
Mode is 'off'. Erroneous 1x links will be activated normally.
-> sm1xLinkMode ignore
-> sm1xLinkMode
Mode is 'ignore'. Erroneous 1x links will not be activated.
```

NOTES: When set to 'off', all links come up normally. When set to 'ignore', links that only come up at 1x (when they were enabled for a higher rate) are forced down. These downed ports can be queried to aid debugging errors in the fabric.

smTrapThreshold

Set/display the urgent trap threshold (in minutes) for port auto-disable.

Syntax:

```
smTrapThreshold [threshold]
```

Options:

None.

Sample Output:

```
-> smTrapThreshold
Trap Threshold is 0 (disabled).

-> smTrapThreshold 20

-> smTrapThreshold
Trap Threshold is 20 traps/minute.
```

NOTES: When enabled, ports generating urgent traps at a rate higher than the threshold will be disabled. This value can range from 10 to 100 traps/minute. Set the value to 0 to disable this feature.

smAppearanceMsgThresh

Set/display the threshold for Appearance & Disappearance messages.

Syntax:

```
smAppearanceMsgThresh [threshold]
```

Options:

None.

Sample Output:

```
-> smAppearanceMsgThresh
Message Threshold is 0 (disabled).

-> smAppearanceMsgThresh 3

-> smAppearanceMsgThresh
Message Threshold is 3 messages/sweep.
```

NOTES: This command sets the threshold for the number of fabric appearance and disappearance log messages that may be logged as NOTICEs per sweep by the SM. A value of zero causes all such messages to be logged at the NOTICE level. A value greater than zero will cause the priority of any subsequent messages to be logged at the INFO priority.

smPmBmStart

Set/display whether the PM and BM will start with the SM.

Syntax:

```
smPmBmStart [enable] [disable]
```

Options:

enable

Enable the start of the PM and BM at SM start-up.

disable

Disable the start of the PM and BM at SM start-up.

Sample Output:

```
-> smPmBmStart
SM is enabled
PM is enabled
BM is enabled
FE is enabled

-> smPmBmStart disable
SM is enabled
PM is disabled
BM is disabled
FE is enabled
```

smShowRemovedPorts

Display ports that have been automatically removed from the fabric.

Syntax:

```
smShowRemovedPorts
```

Options:

None.

Sample Output:

```
-> smShowRemovedPorts
Disabled Ports:
  Node 0x00066a0007000001: SilverStorm 9080 GUID=0x00066a00da000001
Leaf 1, Chip A
    Port 12: Trap Threshold Exceeded
    Port 23: 1x Link Width
```

NOTES: This displays ports that have been removed from the fabric automatically by the SM, such as when a 1x link mode is set to 'ignore' or when a port has exceeded its urgent trap threshold.

Log

logShow

Displays the log file.

Syntax:

```
logShow
```

Options:

None.

Sample Output:

```
-> logshow

W|2006/10/04 20:26:31.176U: Thread "Log" (0x8fdab3b0)
        Log: Unable to Send Trap: 523:Bad:65535
W|2006/10/04 20:26:31.176U: Thread "CPU1" (0x1)
        Fcpi: Target Device 1 (2e7 p1) NPort Id 0x0106d1: Connection
Restored
W|2006/10/04 20:26:31.196U: Thread "Log" (0x8fdab3b0)
        Log: Unable to Send Trap: 523:Bad:65535
W|2006/10/04 20:26:31.216U: Thread "CPU1" (0x1)
        Fcpi: Target Device 2 (b31 p2) NPort Id 0x0106d2: Connection
Restored
W|2006/10/04 20:26:31.236U: Thread "CPU1" (0x1)
        Fcpi: Target Device 3 (d94 p3) NPort Id 0x0106d3: Connection
Restored
W|2006/10/04 20:26:31.246U: Thread "Log" (0x8fdab3b0)
        Log: Unable to Send Trap: 523:Bad:65535
W|2006/10/04 20:26:31.256U: Thread "CPU1" (0x1)
        Fcpi: Target Device 6 (ac6 p6) NPort Id 0x0106d6: Connection
Restored
W|2006/10/04 20:26:31.276U: Thread "Log" (0x8fdab3b0)
        Log: Unable to Send Trap: 523:Bad:65535
W|2006/10/04 20:26:31.326U: Thread "Log" (0x8fdab3b0)
        Log: Unable to Send Trap: 523:Bad:65535
W|2006/10/04 20:28:29.912U: Thread "tTelnetd" (0x8fe143e0)
        Osa: telnetd: connection requested by 192.168.0.107
W|2006/10/04 20:46:26.113U: Thread "tTelnetd" (0x8fe143e0)
        Osa: telnetd: connection requested by 192.168.0.107
W|2006/10/05 19:37:08.727U: Thread "tTelnetd" (0x8fe143e0)
        Osa: telnetd: connection requested by 192.168.0.46
W|2006/10/05 20:19:20.101U: Thread "tTelnetd" (0x8fe143e0)
```

logClear

Clears the log file.

Syntax:

```
logClear [-noprompt]
```

Options:

-noprompt delete all log messages without prompting the user

Sample Output:

```
-> logClear
Ram Log cleared
```

logConfigure

Configures the log settings.

Syntax:

```
logConfigure
```

Options:

None.

Sample Output:

```
-> logConfigure
Type Q or X to exit.
Please enter the number corresponding to what you want to configure.
index : name          : description
-------------------------
  1   : Device        : Logging device. (IE. Ram, syslog, etc)
  2   : Preset        : General log filter.

Select: 1
Configurable devices
index : name     : |D|F|E|A|W|P|C|I|P|N|1|2|3|4|5|
-----------------------------------------------------
  1   : Ram      : |X|X|X|X|X| | |X| |X| | | | | |
  2   : BriefRam : |X|X|X|X|X| | | | |X| | | | | |
  3   : Console  : |X|X|X|X|X|X|X|X|X|X| | | | | |
  4   : Trap     : |X|X|X|X|X| | | | |X| | | | | |
  5   : Syslog   : |X|X|X|X|X|X| | | |X| | | | | |
Type Q or X to exit

Enter the device index you wish to configure: 1
Level: Dump [1]
Level: Fatal [1]
Level: Error [1]
Level: Alarm [1]
Level: Warning [1]
Level: Partial [0]
Level: Config [0] 1
Level: Info [1]
Level: Periodic [0]
Level: Notice [1]
Level: Debug1 [0]
Level: Debug2 [0]
Level: Debug3 [0]
Level: Debug4 [0]
Level: Debug5 [0]
Log device configuration changed
```

NOTES: This is an interactive command to configure log settings. This involves setting which log levels are active.

Definitions:

- Preset: Enable or disable each log level that may be generated on the system. The log presets act as a general filter. For example, if the **Info** logging level is disabled in the presets, no **Info** messages will be shown on any output device, regardless of **Info** log setting for a device.
- Device: Enable a device to display or process log messages of each level.
- Syslog: Configure the syslog host ip address and port.

logResetToDefaults

Restores the log file default settings.

Syntax:

```
logResetToDefaults [-noprompt]
```

Options:

-noprompt
Restore the defaults without prompting the user.

Sample Output:

```
-> logResetToDefaults
Log configuration has been reset
```

logSyslogConfig

Configure the syslog host IP address.

Syntax:

```
logSyslogConfig [-h xxx.xxx.xxx.xxx] [-p xxxx] [-f xx]
[-m x]
```

Options:

-h ip_address
Sets the host IP address in dotted decimal format (xxx.xxx.xxx.xxx).

-p port
The host port number on which the syslog server is listening.

-f facility
The syslog facility to use in the messages.

-m mode
Determines whether the syslog is to be put into a special OEM mode.

Sample Output:

```
-> logSyslogConfig -h 172.26.0.202
Successfully configured the syslog host
```

NOTES: The device can forward its log messages to a syslog host if configured. This command allows a user to configure the host and port to send messages to and the facility to use in the messages. Additional configuration may be necessary to fully configure the log system.

logShowConfig

Display the current log configuration settings.

Syntax:

```
logShowConfig
```

Options:

None.

Sample Output:

```
Log Configuration for Slot 101:
-------------------------------
Configurable devices
index : name     : |D|F|E|A|W|P|C|I|P|N|1|2|3|4|5|
-----------------------------------------------------
  1   : Ram      : |X|X|X|X|X| |X|X| |X| | | | | |
  2   : BriefRam : |X|X|X|X|X| | | | |X| | | | | |
  3   : Console  : |X|X|X|X|X|X|X|X|X|X| | | | | |
  4   : Trap     : |X|X|X|X|X| | | | |X| | | | | |
  5   : Syslog   : |X|X|X|X|X|X| | | |X| | | | | |

Configurable presets
index : name     : state
-------------------------------
  1   : Dump     : Enabled
  2   : Fatal    : Enabled
  3   : Error    : Enabled
  4   : Alarm    : Enabled
  5   : Warning  : Enabled
  6   : Partial  : Enabled
  7   : Config   : Enabled
  8   : Info     : Enabled
  9   : Periodic : Enabled
  15  : Notice   : Enabled
  10  : Debug1   : Disabled
  11  : Debug2   : Disabled
  12  : Debug3   : Disabled
  13  : Debug4   : Disabled
  14  : Debug5   : Disabled
```

logSyslogTest

Test the Syslog configuration.

Syntax:

```
logSyslogTest [severityType]
```

Options:

-e

Send Error severity CSM test message to Syslog.

-w

Send Warning severity CSM test message to Syslog

-n

Send Warning severity CSM test message to Syslog.

Sample Output:

```
-> logSyslogTest -e
Currently configured Syslog host is: 0.0.0.0 port 514 facility 22
Syslog configuration has been tested
```

Key Management

showKeys

Display the license keys for the unit.

Syntax:

```
showKeys
```

Options:

None.

Sample Output:

```
-> showKeys
-------------------------------------------------------
Key number:  1
Key:         FV7P91-2V9H6F-946QS3-3SCEV5-YZMZ5R-S
Description: Subnet Manager License
Status:      Active
```

NOTES: License keys unlock various software features of the product.

addKey

Add a license key.

Syntax:

```
addKey [key]
```

Options:

key
The license key to add.

Sample Output:

```
-> addKey XXxxx-YYYYY-ZZZZZ-11111-222222-3
```

removeKey

Remove a license key.

Syntax:

```
removeKey
```

Options:

None.

Sample Output:

```
-> removeKey
-------------------------------------------------------
Key number:  1
Key:         XX7P91-2V9H6F-946QS3-3SCEV5-YZMZ5R-S
Description: Subnet Manager License
Status:      Active

Please enter the key number you wish to remove:
```

NOTES: After entering this command a list of available keys will be displayed. The keys are shown along with an associated number. At the prompt, enter the number of the key you wish to remove.

lbSwitchInfo

ismPortStats

Displays link error information associated with each switch port. These statistics include errors, dropped packets, discarded packets, and invalid packets.

Syntax:

```
ismPortStats [-clear] [-noprompt] [-cols X] [-port X]
[-leaf X] [-spine X]
```

Options:

-clear

Clears the statistics. Statistics are displayed first, then cleared.

-noprompt

Does not give the user a 'Continue' prompt for each page of display.

-cols X

Sets the number of columns to be displayed per line.

-port X

Specifies a port(s) to display.

-leaf X

Displays all ports for a specific leaf.

-spine X

Displays all ports for a specific spine.

Sample Output:

```
-> ismPortStats
Name         HSSM_BAY-1 HSSM_BAY-2
PhysState         Up       Up
PortState        Act      Act
LinkWidth         4X       4X
LinkSpeed    2.5Gbps  2.5Gbps
SymblErrors        0        0
ErrRecovery        0        0
LinkDowned         0        0
RcvErrors          0        0
RmtPhysErr         0        0
TxDiscards         0        0
InPKeyViol         0        0
OutPKeyViol        0        0
InRawViol          0        0
OutRawViol         0        0
LLIntegrity        0        0
ExcesBufOvr        0        0
```

NOTES: Port statistic descriptions:

- PhysState - The physical state of the port.
- PortState - The state of the link on this port.
- LinkWidth - The currently active link width on this port.
- LinkSpeed - The currently active link speed on this port. Port speed is LinkWidth * LinkSpeed.
- SymblErrors - The number of times a 8B10B encoding violation, or a disparity violation was detected. If multiple errors are detected simultaneously (in more than one lane), the counter only increments by one.
- ErrRecovery - The number of times the link error recovery proccess happened successfully.
- LinkDowned - The number of times the link error recovery process failed.
- RcvErrors - Number of errors received on the port.
- RmtPhysErr - Number of remote physical errors received on the port.
- TxDiscards - Number of port transmit discards.
- InPKeyViol - Number of times PKey inbound invalid.
- OutPKeyViol - Number of times PKey outbound invalid.
- InRawViol - Number of times raw inbound packet discarded.
- OutRawViol - Number of times raw outbound packet was discarded.
- LLIntegrity - Number of local link integrity errors.
- ExcesBufOvr - Number of excessive buffer overrun errors.

- The options -leaf and -spine are only available on platforms with removable leaf and spine modules.

ismPortCounters

Displays a table comparision of transmit, receive and error counters cooresponding to each port of the module.

Syntax:

```
ismPortCounters [-clear] [-active] [-errors]
[-potential] [-noprompt]
```

Options:

-clear

Clears the counters. Counters are first displayed, then cleared.

-active

Displays only the counters for ports in the active state.

-errors

Displays only the counters for ports with receive symbol errors.

-potential

Displays only the counters for ports with active link or width under their maximum supported value.

-noprompt

Does not give the user a 'Continue' prompt for each page of display on switches with more than 24 ports.

Sample Output:

```
-> ismPortCounters -errors
No ports with symbol errors.
```

NOTES: Port counter descriptions:

- Transmit, Packets - The number of packets transmitted by the port, not including flow control packets.
- Transmit, Words - The number of data words transmitted by the port, not including flow control and VCRC data.
- Transmit, Wait - The number of 4ns ticks during which the port had data to transmit but no data was sent either because of insufficient credits or because of lack of arbitration.
- Receive, Packets - The number of data packets received by the port, not including flow control packets.
- Receive, Words - The number of data words received by the port, not including flow control and VCRC data.
- Errors, Rcv - Number of symbol errors received on the port.

- Errors, Rmt - Number of switch relay errors received on the port.

- For MPFD switches, the counters for the cable ports as well as the ports for each line card are shown. For switches with more than 24 ports, the user is prompted to continue the output after each group of 24 ports are displayed. This command is best displayed with a terminal width of at least 120 columns.

ismLinearFwdb

Displays the entries in the linear forwarding table. LIDs and a cooresponding port are shown. A packet addressed to a LID will be forwared to the cooresonding port listed in the displayed table.

Syntax:

```
ismLinearFwdb [switch]
```

Options:

switch
switch number.

Sample Output:

```
-> ismLinearFwdb
Switch Leaf 3 Linear Fwdb:
 LID :: Port
0001      19 (L03S1Aa)
0002      0 ( )
0003      12 (L03P12 )
0004      13 (L03S1Ba)

Switch Spine 1-A Linear Fwdb:
 LID :: Port
0001      0 ( )
0002      9 (S1AL03a)
0003      9 (S1AL03a)
0004      11 (S1AL03b)
```

NOTES: The 'set' option is not available on the SilverStorm 9000 products.

ismMultiFwdb

Show Multicast Forwarding Database for switch.

Syntax:

```
ismMultiFwdb [switch]
```

Options:

switch
Switch identifier.

Sample Output:

```
-> ismMultiFwdb
Switch Leaf 3 Multicast Fwdb:
c000      12 (L03P12 ) 19 (L03S1Aa)
c001      12 (L03P12 ) 19 (L03S1Aa)

Switch Spine 1-A Multicast Fwdb:
c000       9 (S1AL03a)
c001       9 (S1AL03a)
```

NOTES: This command is best displayed with a terminal width of at least 120 columns.

ismAutoClearConf

This feature will clear the InfiniBand port statistic counters that have reached their maximum.

Syntax:

```
ismAutoClearConf [disable | enable [warn] [log_first]]
```

Options:

enable

Enable the auto-clear feature.

disable

Disable the auto-clear feature.

warn

Generate warning log messages instead of the default information messages.

log_first

Log first clear (otherwise the first clear is not logged).

Sample Output:

```
-> ismAutoClearConf enable
Auto clear is enabled
```

NOTE: This feature will log every time a counter has reached its maximum capacity. This may be useful for diagnostics purposes, specifically for bad cables. This feature is only available on certain switch hardware platforms.

ismPortSet12x

Allows the user to view, set and unset port link width to 12X for specific port(s).

Syntax:

```
ismPortSet12x portName [enable12xBit]
```

Options:

portName

A valid 6 or 7 character port name (e.g., S1AL01a or L01P01).

enable12xBit

2=enable(12X only) 1=enable(AUTO 1X/4X/12X), 0=disable.

Sample Output:

```
-> ismPortSet12x S1AL01a
S1AL01a 12x mode is DISABLED
```

NOTES: This command works only on platforms supporting 12X link aggregation.

The option portName is case sensitive. If it contains spaces, it must be bounded by quotes (e.g., ismPortSet12x "Cable 1", 1). Using this command without the option enable12xBit displays the current values of the port.

Devices with effected ports MUST BE REBOOTED to activate changes made with ismPortSet12x.

ismChassisSet12x

Allows the user to view, set and unset the chassis link width to 12X (including external ports).

Syntax:

```
ismChassisSet12x [enable12xBit]
```

Options:

enable12xBit

2=enable(12X only) 1=enable(AUTO 1X/4X/12X), 0=disable.

Sample Output:

```
-> ismChassisSet12x
L01P01  12x mode is DISABLED
L01P02  12x mode is DISABLED
L01P03  12x mode is DISABLED
L01P04  12x mode is DISABLED
L01P05  12x mode is DISABLED
L01P06  12x mode is DISABLED
L01P07  12x mode is DISABLED
L01P08  12x mode is DISABLED
L01P09  12x mode is DISABLED
L01P10  12x mode is DISABLED
L01P11  12x mode is DISABLED
```

NOTES: This command works only on platforms supporting 12X link aggregation (currently the 9024 and 9080).

This command will enable 12X link aggregation so that internal and external switching will be 12x. Each 12X port is 3 aggregated 4x ports (i.e., a TRIO), where one will be reported as a 12x port, and the other two ports are reported as disabled. Each of the three aggregated 4x ports must be connected to another enabled 12X TRIO to establish a 12X link. Using this command without the option enable12xBit displays the ports current values. The device must be rebooted to activate changes made with this command.

Calling this command without the enable12xBit will display the current values of the port(s).

ismChassisSetSpeed

Allows the user to view and set port link speeds (including external ports) for the entire chassis to 2.5 (SDR), 5.0 (DDR) or AutoNegotiate.

Syntax:

```
ismChassisSetSpeed [speed]
```

Options:

speed

1=2.5, 2=5.0 or 3=AutoNegotiate.

Sample Output:

```
-> ismChassisSetSpeed
L01P01  link speed is AUTO NEGOTIATE
L01P02  link speed is AUTO NEGOTIATE
L01P03  link speed is AUTO NEGOTIATE
L01P04  link speed is AUTO NEGOTIATE
L01P05  link speed is AUTO NEGOTIATE
L01P06  link speed is AUTO NEGOTIATE
L01P07  link speed is AUTO NEGOTIATE
L01P08  link speed is AUTO NEGOTIATE
```

NOTE: DDR links (5.0) are only available on chassis supporting DDR. This command sets link speeds for both internal and external ports .Each external DDR port must be connected to another DDR port to establish a 5.0 link. Using this function without the speed option displays the current values of the port(s).A reboot of the chassis is required to activate changes made with this command.

ismPortEnable

Allows the user to enable or disable a port.

Syntax:

```
ismPortEnable port
```

Options:

port
Port name.

Sample Output:

```
-> ismPortEnable S3BL08b
```

NOTE: The option portName is case sensitive. If it contains spaces, it must be bounded by quotes (e.g., ismPortEnable "Cable 1"). A reboot is required to activate changes made with this command.

ismChassisSetEnable

Allows the user to view and enable ports for entire chassis.

Syntax:

```
ismChassisSetEnable [enable]
```

Options:

enable
1=enable, 0=disable.

Sample Output:

```
-> ismChassisSetEnable
L01P01  is ENABLED
L01P02  is ENABLED
L01P03  is ENABLED
L01P04  is ENABLED
L01P05  is ENABLED
L01P06  is ENABLED
L01P07  is ENABLED.
```

NOTE: Using this command without the enable option displays current value of the port(s). A reboot is required to activate changes made with this command.

ismChassisSetDdrPreemphasis

Allows the user to view and set the preemphasis for all cable side DDR ports.

Syntax:

```
ismChassisSetDdrPreemphasis [preemphasis]
```

Options:

preemphasis

port preemphasis (4 bits per SERDES on the port. Use the format 0x01010101, substituting for the 1(s).

Sample Output:

```
-> ismChassisSetDdrPreemphasis
Cable 1 DDR preemphasis=0x01010101
Cable 2 DDR preemphasis=0x01010101
Cable 3 DDR preemphasis=0x01010101
Cable 4 DDR preemphasis=0x01010101
Cable 5 DDR preemphasis=0x01010101
Cable 6 DDR preemphasis=0x01010101
Cable 7 DDR preemphasis=0x01010101
Cable 8 DDR preemphasis=0x01010101
Cable 9 DDR preemphasis=0x01010101
Cable10 DDR preemphasis=0x01010101
Cable11 DDR preemphasis=0x01010101
Cable12 DDR preemphasis=0x01010101
Cable13 DDR preemphasis=0x01010101
Cable14 DDR preemphasis=0x01010101
Cable15 DDR preemphasis=0x01010101
Cable16 DDR preemphasis=0x01010101
Cable17 DDR preemphasis=0x01010101
Cable18 DDR preemphasis=0x01010101
Cable19 DDR preemphasis=0x01010101
Cable20 DDR preemphasis=0x01010101
Cable21 DDR preemphasis=0x01010101
Cable22 DDR preemphasis=0x01010101
Cable23 DDR preemphasis=0x01010101
Cable24 DDR preemphasis=0x01010101
```

NOTE: Calling this function with no parameters will display the current value for each port. A reboot is required to activate changes made with this command.

ismPortDisable

Allows the user to view and disable ports for entire chassis.

Syntax:

```
ismPortDisable port
```

Options:

port
Port name.

Sample Output:

```
-> ismPortDisable S3BL10a
```

NOTE: The option port is case sensitive. If it contains spaces, it must be bounded by quotes (e.g., ismPortEnable "Cable 1"). A reboot is required to activate changes made with this command.

ismPortSetSpeed

Allows the user to view and set port speeds.

Syntax:

```
ismPortSetSpeed port [speed]
```

Options:

port
Port name.

speed
The port speed (1=SDR, 2=DDR, 3=AutoNegotiate).

Sample Output:

```
-> ismPortSetSpeed S3BL08b
S3BL08b link speed is AUTO NEGOTIATE.
```

NOTE: The option port is case sensitive. If it contains spaces, it must be bounded by quotes (e.g., ismPortSetSpeed "Cable 1"). Using this command with only the port option displays its current values. A reboot is required to activate changes made with this command.

ismPortSetDdrPreemphasis

Allows the user to view and set preemphasis for a cable side DDR port.

Syntax:

```
ismPortSetDdrPreemphasis [port] [preemphasis]
```

Options:

port

Port name.

preemphasis

port preemphasis (4 bits per SERDES on the port, use the format 0x01010101, substitute for the 1(s))

Sample Output:

```
-> ismPortSetDdrPreemphasis L08P12 0x01010101

Devices with effected ports MUST BE REBOOTED to activate changes made with
ismPortSetDdrPreemphasis
```

NOTE: The option portName is case sensitive and if it contains spaces, it must be bounded by quotes. For example, ismPortSetPreemphasis "Cable 1", 1. Calling this function with only the option portName will display the current values.. A reboot is required to activate changes made with this command.

ismIslSet12x

Allows the user to view, set and unset the chassis link width to 12X (except external ports).

Syntax:

```
ismIslSet12x [enable12xBit]
```

Options:

enable12xBit

2=enable(12X only), 1=enable, 0=disable.

Sample Output:

```
-> ismIslSet12x
L01S2Ba 12x mode is DISABLED
L01S2Bb 12x mode is DISABLED
L01S2Bc 12x mode is DISABLED
L01S1Ba 12x mode is DISABLED
L01S1Bb 12x mode is DISABLED
L01S1Bc 12x mode is DISABLED
L01S2Aa 12x mode is DISABLED
L01S2Ab 12x mode is DISABLED
```

NOTE: This command works only on chassis supporting 12X link aggregation. This command enables 12X link aggregation so that internal switching will be 12X. External ports remain configured to 4X and do not require special cabling configuration to function correctly. Using this function without the option enable12xBit displays current values of a port. A reboot is required to activate changes made with this command.

ismChassisSetMtu

Allows the user to view, set and unset the chassis maximum packet MTU Capability and VL Capability for all ports.

Syntax:

```
ismChassisSetMtu [mtuCap] {vlCap}
```

Options:

mtuCap

4=2048 bytes, 5=4096 bytes.

vlCap

1=VL0, 2=VL0,VL1, 3=VL0-VL3, 4=VL0-VL7

Sample Output:

```
-> 1-> ismChassisSetMtu
L01P01  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
L01P02  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
L01P03  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
L01P04  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
L01P05  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
L01P06  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
L01P07  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
L01P08  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
L01P09  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
L01P10  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
L01P11  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
L01P12  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
```

NOTE: This command will set the MTU capability for both internal and external switching. If the vlCap option is not specified, the command defaults to the maximumVL(s) for the selected mtuCap. Calling this function without an mtuCap designator displays the current value for each port. A reboot is required to activate changes made with this command.

ismIslSetSpeed

Allows the user to view and set internal port link speeds to 2.5 (SDR), 5.0 (DDR) or AutoNegotiate.

Syntax:

```
ismIslSetSpeed [speed]
```

Options:

speed

1 = 2.5, 2 = 5.0, 3 = AutoNegotiate.

Sample Output:

```
-> ismIslSetSpeed
L01S3Ba link speed is AUTO NEGOTIATE
L01S1Ba link speed is AUTO NEGOTIATE
L01S1Bb link speed is AUTO NEGOTIATE
L03S1Ba link speed is fixed 2.5 (SDR only device)
L03S3Ba link speed is fixed 2.5 (SDR only device)
```

NOTE: This command works only on chassis supporting DDR links. This command sets inter-switch link speeds to either 2.5, 5.0, or AutoNegotiate. External ports remain configured to SDR rate (2.5) and do not require special cabling configuration to function correctly. Using this command without the option enable12xBit displays the current values of the port. A reboot is required to activate changes made with this command.

ismShowPStatThresh

Displays the port statistic thresholds.

Syntax:

```
ismShowPStatThresh
```

Options:

None.

Sample Output:

```
-> ismShowPStatThresh
        Field              Threshold  Time Unit
        -----              ---------  ---------
 1           portXmitData:         0  (Percent of Max)
 2           portRecvData:         0  (Percent of Max)
 3           portXmitPkts:         0  (Percent of Max)
 4           portRecvPkts:         0  (Percent of Max)
 5           portXmitWait:         0  (1 Second)
 6         portSymbolErr:          0  (1 Second)
 7       portLinkErrRecv:          0  (1 Second)
 8        portLinkDowned:          0  (1 Second)
 9           portRecvErr:          0  (1 Second)
10    portRecvRemPhysErr:          0  (1 Second)
12       portXmitDiscard:          0  (1 Second)
13        portPKeyViolIn:          0  (1 Second)
14       portPKeyViolOut:          0  (1 Second)
15         portRawViolIn:          0  (1 Second)
16        portRawViolOut:          0  (1 Second)
17    portLocalLinkInteg:          0  (1 Second)
18  portExcBufferOverrun:          0  (1 Second)
19 portRelayedVL15Dropped:         0  (1 Second)
20  portLocalVL15Dropped:          0  (1 Second)
21     portNonSMPDropped:          0  (1 Second)
```

NOTE: The following are the available port statics thresholds:

- 1 portXmitData: 32-bit data words transmitted
- 2 portRecvData: 32-bit data words received
- 3 portXmitPkts: data packets transmitted
- 4 portRecvPkts: data packets received
- 6 portSymbolErr: a 8B10B encoding violation, or a displarity violation was detected
- 7 portLinkErrRecv: link error recovery process happened successfully
- 8 portLinkDowned: link error recovery process failed
- 9 portRecvErr: errors received
- 10 portRecvRemPhysErr: remote physical errors received
- 12 portXmitDiscard: port transmit discards
- 13 portPKeyViolIn: PKey inbound was invalid
- 14 portPKeyViolOut: PKey outbound was invalid
- 15 portRawViolIn: raw inbound packet discarded
- 16 portRawViolOut: raw outbound packet discarded
- 17 portLocalLinkInteg: link integrity errors

- 18 portExcBufferOverrun: excessive buffer overrun errors
- 19 portRelayedVL15Dropped: remote VL15 packet was dropped
- 20 portLocalVL15Dropped: local VL15 packet was dropped
- 21 portNonSMPDropped: non SMP packet was dropped

ismSetPStatThresh

Modifies the port statistic thresholds.

Syntax:

```
ismSetPStatThresh field threshold
```

Options:

-field

The name of the port status threshold field.

-threshold

The numeric threshold value.

Sample Output:

```
-> ismSetPStatThresh  portXmitData 0
successfully set port stat threshold portXmitData
```

NOTE: The available thresholds are:

- portXmitData
- portRecvData
- portXmitPkts
- portRecvPkts
- portSymbolErr
- portLinkErrRecv
- portLinkDowned
- portRecvErr
- portRecvRemPhysErr
- portXmitDiscard
- portPKeyViolIn
- portPKeyViolOut
- portRawViolIn

- portRawViolOut
- portLocalLinkInteg
- portExcBufferOverrun
- portRelayedVL15Dropped
- portLocalVL15Dropped
- portNonSMPDropped

ismRemoveStateDump

Removes switch ASIC state dumps files.

Syntax:

```
ismRemoveStateDump [all] [-leaf X] [-spine X]
```

Options:

all

Removes all switch ASIC state dump files.

-leaf X

Removes a switch ASIC state dump file for a specific leaf X.

-spine X

Removes the switch ASIC state dump files for a specific spine X.

Sample Output:

```
-> ismRemoveStateDump
```

ismShowStateDump

Shows contents of switch ASIC state dumps.

Syntax:

```
ismShowStateDump [-full] [all] [-leaf X] [-spine X]
```

Options:

-full

Shows the full contents of the switch ASIC state dump file. Without this option, just the header is displayed.

all

Removes all switch ASIC state dump files.

-leaf X

Removes a switch ASIC state dump file for a specific leaf X.

-spine X

Removes the switch ASIC state dump files for a specific spine X.

Sample Output:

```
-> ismRemoveStateDump
```

TimeManagement

time

Configure the time on the device.

Syntax:

```
time -S ipaddr | -T hhmmss[mmddyyyy]
```

Options:

-S ipaddr

Sets the NTP Server IP address.

-T hhmmss[mmddyyyy]

Set the local clock time hour, minutes, and seconds along with the month, day, and year.

Sample Output:

```
-> time -S 172.26.0.254
Configured the NTP server ip address successfully
13:53:02  10/06/2006
Configured to use NTP server IP address: 172.26.0.254
```

NOTE: Time can be configured locally (using a local clock) or set to be updated by an SNTP server. Please note that if you set the time locally, the unit will unconfigure the NTP server IP address if set. If no options are passed in, the current system time will be printed out.

timeZoneConf

Display/configure the time zone setting.

Syntax:

```
timeZoneConf [offset]
```

Options:

offset

The time offset in relation to Greenwich Mean Time (GMT).

Sample Output:

```
-> timeZoneConf -5
Timezone offset successfully configured
Current time zone offset is: -5
```

NOTE: The offset parameter specifies a time zone the system should use when setting the time. In the U.S. the following time zones are in effect: Eastern Standard Time = GMT -5; Central Standard Time = GMT -6; Mountain Standard Time = GMT -7; Pacific Standard Time = GMT -8. GMT = GreenwichMean Time..

timeDSTConf

Configure and display the Daylight Saving Time settings.

Syntax:

```
timeDSTConf [sw sd sm ew ed em]
```

Options:

sw

Start which, valid values: 1 = 1st, 2 = 2nd, 3 = 3rd, 4 = 4th, 5 = 5th

sd

Start day, valid values: 1 = Sunday, 2 = Monday, 3 = Tuesday, 4 = Wednesday, 5 = Thursday, 6 = Friday, 7 = Saturday

sm

Start month, valid values: 3 = March, 4 = April, 5 = May, 6 = June, 7 = July, 8 = August, 9 = September, 10 = October, 11 = November

ew

End which, valid values: 1 = 1st, 2 = 2nd, 3 = 3rd, 4 = 4th, 5 = 5th

ed

End day, valid values: 1 = Sunday, 2 = Monday, 3 = Tuesday, 4 = Wednesday, 5 = Thursday, 6 = Friday, 7 = Saturday

em

End month, valid values: 3 = March, 4 = April, 5 = May, 6 = June, 7 = July, 8 = August, 9 = September, 10 = October, 11 = November

Sample Output:

```
-> timeDSTConf 2 1 3 1 1 11
Timezone offset successfully configured
Current DST = Start: 2'nd Sunday of March End: 1'st Sunday of November
```

NOTE: Example: To set the daylight savings time to start on the 1st Sunday of April and end on the 4th Sunday of October the command would be: 'timeDSTConf 1 1 4 4 1 10'.

timeNtpTimeout

Display or set the number of seconds to wait for a NTP response.

Syntax:

```
timeNtpTimeout numSeconds
```

Options:

numSeconds
New timeout setting.

Sample Output:

```
-> timeNtpTimeout
Current NTP timeout value: 2 seconds
-> timeNtpTimeout 3
Current NTP timeout changed to 3 seconds
```

NOTE: With no arguments, this command displays the current NTP timeout settings. This is the amount of time (in seconds) for the system to wait for a response from the NTP server. This setting can be configured by using the same command with the new timeout value (in whole seconds) as the only argument. The default settings is 2 seconds. The NTP timeout value is not used on linecards or slave CMUs.

timeNtpRefreshTime

Display or set the delay between syncing the clock via NTP.

Syntax:

```
timeNtpRefreshTime numSeconds
```

Options:

numSeconds
New refresh delay setting.

Sample Output:

```
-> timeNtpRefreshTime
Current NTP refresh delay value: 60 seconds
-> timeNtpRefreshTime 50
Current NTP refresh delay changed to 50 seconds.
```

NOTE: With no arguments this command displays the current NTP refresh time, which is the delay in seconds between attempts to sync the clock via NTP. This value can be configured by using this same

command with the new refresh time (in whole seconds) as the only argument. The NTP refresh time is not used on linecards or slave CMUs.

SNMP

snmpCommunityConf

Displays and allows the changing of the SNMP community strings.

Syntax:

```
snmpCommunityConf [-r readonly_comm_str]
[-w read_write_comm_str]
```

Options:

-r

A read-only community string.

-w

A read/write community string.

Sample Output:

```
-> snmpCommunityConf -r public
Read Only Community String Was Set To: public
```

NOTE: If no options are entered the current SNMP read-only community string and read-write community string is displayed. If options are entered then the corresponding community string is changed.

snmpTargetAddr

Displays and allows the changing of the **snmpTargetAddrTable** entries.

Syntax:

```
snmpTargetAddr [show | delete | add | edit] -n name
[-a addr] [-p port] [-t timeout] [-r retry_count]
[-l tag_list] [-v parameters] [-s storage_type]
```

Options:

show

Displays the contents of the **snmpTargetAddrTable**.

add

Adds a row to the **snmpTargetAddrTable**.

edit

Modifies an existing row in the **snmpTargetAddrTable**.

delete

Removes an existing row of the **snmpTargetAddrTable**.

-n

Name. A unique name used to identify a row. Any name with a space (e.g. xxx v3) must be surrounded by double quotes (").

-a

Address. The target machine IP address in dotted decimal form.

-p

Port. The target port to send traps and information.

-t

Timeout. The time to wait for an information response.

-r

Retry count. The number of re-send attempts for information.

-l

Tag list. Indicates the traps and information that is sent.

-v

Parameters. This maps to an entry in the **snmpTargetAddrTable.**

-s

Storage type. Determines whether the entry is saved in flash memory.

-i

Status, 1=Active, 2=Not In Service, 3=Not Ready.

Sample Output:

```
-> snmpTargetAddr
rfc2573t:snmpTargetAddrTDomain: nms v1 : 1.3.6.1.6.1.1
rfc2573t:snmpTargetAddrTDomain: nms v2 : 1.3.6.1.6.1.1
rfc2573t:snmpTargetAddrTDomain: nms v3 : 1.3.6.1.6.1.1
rfc2573t:snmpTargetAddrTAddress: nms v1 :  (ip addr)00.00.00.00 (port)0000
rfc2573t:snmpTargetAddrTAddress: nms v2 :  (ip addr)00.00.00.00 (port)0000
rfc2573t:snmpTargetAddrTAddress: nms v3 :  (ip addr)00.00.00.00 (port)0000
```

NOTE: The output is in the form: 'mib : mib_object : table_index : value' . For more details on the snmpTargetAddrTable see SNMP-TARGET-MIB, RFC 2573.

snmpTargetParams

Displays the **snmpTargetParamsTable** entries.

Syntax:

```
snmpTargetParams [show]
```

Options:

show
Displays the contents of the **snmpTargetParamsTable**.

Sample Output:

```
-> snmpTargetParams
rfc2573t:snmpTargetParamsMPModel: v1 params : 0
rfc2573t:snmpTargetParamsMPModel: v2 params : 1
rfc2573t:snmpTargetParamsMPModel: v3 params : 3
rfc2573t:snmpTargetParamsSecurityModel: v1 params : 1
rfc2573t:snmpTargetParamsSecurityModel: v2 params : 2
rfc2573t:snmpTargetParamsSecurityModel: v3 params : 3
```

NOTE: The output is in the form: 'mib : mib_object : table_index : value' . For more details on the snmpTargetParamsTable see SNMP-TARGET-MIB, RFC-2573.

snmpNotifyProfile

Displays the **snmpNotifyFilterProfileTable** entries.

Syntax:

```
snmpNotifyProfile [show]
```

Options:

show
Displays the contents of the **snmpNotifyFilterProfileTable**.

Sample Output:

```
-> snmpNotifyProfile
rfc2573n:snmpNotifyFilterProfileName: v1 params : v1 params
rfc2573n:snmpNotifyFilterProfileName: v2 params : v2 params
rfc2573n:snmpNotifyFilterProfileName: v3 params : v3 params
rfc2573n:snmpNotifyFilterProfileStorType: v1 params : 3
rfc2573n:snmpNotifyFilterProfileStorType: v2 params : 3
rfc2573n:snmpNotifyFilterProfileStorType: v3 params : 3
```

NOTE: The output is in the form: 'mib : mib_object : table_index : value' . For more details on the snmpNotifyFilterProfileTable see SNMP-NOTIFICATION-MIB, RFC-2573.

snmpNotifyFilter

Displays the **snmpNotifyFilterTable** entries.

Syntax:

```
snmpNotifyFilter [show]
```

Options:

show
Displays the contents of the **snmpNotifyFilterTable**.

Sample Output:

```
-> snmpNotifyFilter
rfc2573n:snmpNotifyFilterMask:  v1 params : 0
rfc2573n:snmpNotifyFilterMask:  v2 params : 0
rfc2573n:snmpNotifyFilterMask:  v3 params : 0
rfc2573n:snmpNotifyFilterType:  v1 params : 1
rfc2573n:snmpNotifyFilterType:  v2 params : 1
rfc2573n:snmpNotifyFilterType:  v3 params : 1
```

NOTE: The output is in the form: 'mib : mib_object : table_index : value' . For more details on the snmpNotifyFilterTable see SNMP-NOTIFICATION-MIB, RFC-2573.

snmpNotify

Displays the **snmpNotifyTable** entries.

Syntax:

```
snmpNotify [show]
```

Options:

show
Displalys the contents of the **snmpNotifyTable**.

Sample Output:

```
-> snmpNotify
rfc2573n:snmpNotifyTag: switch : rfc1493
rfc2573n:snmpNotifyTag: interfaces : rfc2233
rfc2573n:snmpNotifyTag: rmon : rfc1757
rfc2573n:snmpNotifyTag: snmp : rfc1907
rfc2573n:snmpNotifyTag: tms : tmscom
rfc2573n:snmpNotifyType: switch : 1
```

NOTE: The output is in the form: 'mib : mib_object : table_index : value' . For more details on the snmpNotifyTable see SNMP-NOTIFICATION-MIB, RFC-2573.

snmpSystem

Displays and allows the changing of the Mib-2 system entries.

Syntax:

```
snmpSystem [show | edit] [-n sysName] [-l sysLocation]
[-c sysContact]
```

Options:

show
show the contents of the snmpTargetAddrTable.

edit
modify an existing row in the snmpTargetAddrTable.

-n
sysName.

-c
sysContact.

-l
sysLocation.

Sample Output:

```
-> snmpSystem show
rfc1907:sysDescr:  : BC2FC
rfc1907:sysObjectID:  : 1.3.6.1.4.1.10222.7.2.2
rfc1907:sysUpTime:  : 0 Day(s), 0 Hour(s), 28 Minute(s), 28 Second(s)
rfc1907:sysContact:  :
2D.2D.45.6D.70.74.79.3B.20.4E.6F.20.56.61.6C.75.65.20.53.65.74.2D.2D
rfc1907:sysName:  : BC2FC-4
rfc1907:sysLocation:  : Main Chassis Unit, Slot 4
rfc1907:sysServices:  : 79
```

NOTE: The output is in the form: 'mib : mib_object : table_index : value'.

snmpUsrSec

Configure and display SNMP V3 users.

Syntax:

```
snmpUsrSec [add | show | edit | delete] [name] [-a algo
[key]]
```

Options:

add

add an entry to the V3 user table.

show

show entries in the V3 user table.

edit

modify an entry in the V3 user table.

delete

remove an entry in the V3 user table.

-a

authentication algorithm and key. Options: MD5, SHA, NONE. Passphrase is required unless NONE is specified.

Sample Output:

```
-> snmpUsrSec
User    : initialmd5
Auth    : MD5
Auth Key: 0x047b473f93211a17813ce5fff290066b
Priv    : NONE

User    : initialsha
Auth    : SHA
Auth Key: 0x1c8cbd687fb0f0a22ddd24315db0d84c09eb5ff3
Priv    : NONE

User    : initialnone
Auth    : NONE
Priv    : NONE
```

NOTE: Handles configuration and display of SNMP v3 users. Supported authentication algorithms are: NONE, MD5, and SHA. A passphrase is required for all except the NONE algorithm. No privacy algorithms are currently supported.

Capture

capture

Display information for this device.

Syntax:

```
capture
```

Options:

None.

NOTE: This command is intended for support personnel.

captureFw

Display firmware information for this device.

Syntax:

```
captureFw
```

Options:

None.

NOTE: This command is intended for support personnel.

captureLog

Display log information for this device.

Syntax:

```
captureLog
```

Options:

None.

NOTE: This command is intended for support personnel.

captureSm

Display Subnet Management information for this device.

Syntax:

```
captureSm
```

Options:

None.

NOTE: This command is intended for support personnel.

captureIsm

Display Infiniband switch information for this device.

Syntax:

```
captureIsm
```

Options:

None.

NOTE: This command is intended for support personnel.

captureChassis

Display chassis information for this device.

Syntax:

```
captureChassis
```

Options:

None.

NOTE: This command is intended for support personnel.

captureNetwork

Display chassis information for this device.

Syntax:

```
captureNetwork
```

Options:

None.

NOTE: This command is intended for support personnel.

captureMisc

Display miscellaneous information for this device.

Syntax:

```
captureMisc
```

Options:

None.

NOTE: This command is intended for support personnel.

captureSnmp

Display SNMP information for this device.

Syntax:

```
captureSnmp
```

Options:

None.

NOTE: This command is intended for support personnel.

captureShell

Display shell command information for this device.

Syntax:

```
captureShell
```

Options:

None.

NOTE: This command is intended for support personnel.

3 Virtual I/O Command Line Interface

Overview

This section details the usage of the Command Line Interface (CLI) feature for the EVIC and FVIC virtual I/O (VIO) modules. The CLI allows the user to perform remote configuration and management tasks, which in many respects mirrors the functionality of the Chassis Viewer GUI.

The VIO CLI is accessed via the **`rlogin`** command executed from the 9000 switch CLI.

Commands and Functional Groups

The list of available commands can be accessed by typing **list**. To keep the list short, the commands are grouped into functional groups, which are:

General:

General administrative commands

Deprecated:

Commands that have been deprecated.

The Deprecated group contains CLI commands that have been replaced or are to be removed. Please use the new command where appropriate.

Linecard:

Commands for linecard (EVIC and FVIC) information.

Firmware:

Provides commands for updating the firmware via a File Transfer Protocol (FTP) server or Secure Copy Protocol (SCP) (if using SSH to access the Bridge Module). The switch has the ability to store the location of the firmware files for future upgrades. Additionally, the Firmware functional group includes commands for viewing the current firmware revisions and for changing the boot image.

SubnetManagement:

InfiniBand subnet manager configuration and management.

NOTE: For 9020 users, the subnet manager CLI commands are accessed via the VIO modules.

Log:

Provides commands for viewing log files as well as configuring logging parameters.

KeyManagement:

License key management.

NOTE: For 9020 users, the Key Management CLI commands are accessed via the VIO modules.

Capture:

Provides commands for capturing switch-specific information for the purposes of analysis and debugging.

FibreChannel:

Specific to the FVIC, provides commands for managing the Fibre Channel interface.

Ethernet:

Specific to the EVIC, provides commands for managing the Fibre Channel interface.

Configuration:

Provides commands for loading, saving and displaying configuration files.

To list commands within a functional group, simply type in the functional group name. For example, to list all of the firmware commands, type **Firmware**. The system would display the following:

```
-> Firmware
fwVersion           Display Firmware revisions
bootQuery           Query boot image information
bootSelect          Change boot selection
```

Online Help

The online help for the CLI provides, for each command, all necessary information to successfully execute the command. For example, typing **help fwVersion** displays the following information for the firmware update command:

```
NAME
       fwVersion
SYNOPSIS
       fwVersion
DESCRIPTION
       Displays the firmware versions for the unit.
OPTIONS
       None.
```

Keyboard Shortcuts

- The CLI keeps a history of recently executed commands. This history is available via the **Up** and **Down** arrow keys.
- Users may edit the current command with the **Left** and **Right** arrow keys.
- Tab completion: pressing the **Tab** key after typing at least one character either completes a command or lists all the available commands that begin with the characters already typed.

Accessing the VIO CLI

NOTE: The access the VIO CLI the user must first log into the CLI of the applicable 9000 switch.

1. Telnet to the IP address of the switch (the default IP address is 192.168.100.9) with the following command:

   ```
   open <IP ADDRESS>
   ```

2. The system prompts for a username. The CLI has the following default user names:

   Operator access: **operator**

   Administrator access: **admin**

   Type the appropriate username and press **Enter**.

3. The system prompts for a password. The CLI has the following default passwords:

   Operator access: **operpass**

Administrator access: **adminpass**

Type the appropriate password and press **Enter**. The system responds with:

```
Welcome to the <SWITCH> CLI. Type 'list' for the list of
commands.
```

4. At the system prompt enter the following command::

```
rlogin <slot#>
```

where <slot#> corresponds to the switch slot where the VIO card resides. The system responds with:

```
Welcome to the  CLI. Type 'list' for the list of commands..
```

Groups and Commands

The following section lists the CLI's functional groups along with the commands for each group. For more specific information for each functional group, the user would execute the **help <GROUP NAME>** command. For more specific command information, the user would execute the **help <COMMAND NAME>** command.

NOTE: This section lists all known CLI commands. The CLI commands for a particular product may vary.

General

NOTE: For detailed information on commands for the General group, refer to "General" on page 2-5.

Linecard

cardInfo

Shows Linecard Information.

Syntax:

```
cardInfo
```

Options:

None.

Sample Output:

```
-> cardInfo
Out of Band LAN IP  : 172.26.2.87
Net Mask            : 255.255.240.0
MAC Address         : 00:06:6a:00:50:ba
Description         : --Empty; No Value Set--
Alias               : VIO
IB Node Description : VIO, slot L6, 0x133
Non-Fatal Errors    : 0
Fatal Errors        : 0
Card Description    : VIO
Card Uptime         : 0 Day(s), 0 Hour(s), 9 Minute(s), 3 Second(s)
Card Contact        : --Empty; No Value Set--
Card Name           : VIO
Card Location       : Main Chassis Unit, Slot 6
```

NOTE: cardInfo is a related command.

cardNonFatalErrGet

Displays linecard non-fatal error count.

Syntax:

```
cardNonFatalErrGet
```

Options:

None.

Sample Output:

```
-> cardNonFatalErrGet
Non-Fatal Errors    : 0
```

NOTE: cardInfo is a related command.

cardFatalErrGet

Displays linecard fatal error count.

Syntax:

```
cardFatalErrGet
```

Options:

None.

Sample Output:

```
-> cardFatalErrGet
Fatal Errors    : 0
```

NOTE: cardInfo is a related command.

cardUptimeGet

Displays linecard uptime.

Syntax:

```
cardUptimeGet
```

Options:

None.

Sample Output:

```
-> cardUptimeGet
Card Uptime         : 0 Day(s), 0 Hour(s), 12 Minute(s), 36 Second(s)
```

NOTE: cardInfo is a related command.

cardDescGet

Displays linecard description.

Syntax:

```
cardDescGet
```

Options:

None.

Sample Output:

```
-> cardDescGet
Description         : --Empty; No Value Set--
```

NOTE: cardInfo is a related command.

cardDescSet

Modifies the linecard description.

Syntax:

```
cardDescSet <string>
```

Options:

string

Decription text string.

Sample Output:

```
-> cardDescSet "EVIC"
Description         : EVIC
```

NOTE: cardInfo is a related command.

cardAliasGet

Displays the linecard alias.

Syntax:

```
cardAliasGet
```

Options:

None.

Sample Output:

```
-> cardAliasGet
Alias               : FVIC-6
```

NOTE: cardInfo is a related command.

cardAliasSet

Modifies the linecard Alias.

Syntax:

```
cardAliasSet <string>
```

Options:

string
Decription text string.

Sample Output:

```
-> cardAliasSet "EVIC-4"
Alias               : EVIC-4
```

NOTE: cardInfo is a related command.

cardIbNodeDescGet

Displays the linecard IB node description.

Syntax:

```
cardIbNodeDescGet [<default>]
```

Options:

default
Default string.

Sample Output:

```
-> cardIbNodeDescGet default
IB Node Description : EVIC in Chassis 0x0000000000000000, Slot 6
```

NOTE: cardInfo is a related command.

cardIbNodeDescSet

Modifies the linecard IB Node Description.

Syntax:

```
cardIbNodeDescSet <default> <string>
```

Options:

default

Default string.

string

Description text string.

Sample Output:

```
-> cardIbNodeDescSet "EVIC, slot 4, 0x133"
IB Node Description : EVIC, slot 4, 0x133
```

NOTE: cardInfo is a related command.

cardCardDescGet

Shows Linecard "card" description.

Syntax:

```
cardCardDescGet
```

Options:

None.

Sample Output:

```
-> cardCardDescGet
```

NOTE: cardInfo is a related command.

cardCardDescSet

Modifies the linecard IB Node Description.

Syntax:

```
cardCardDescSet <string>
```

Options:

string

Description text string.

Sample Output:

```
-> cardCardDescSet "EVIC, slot 4, 0x133"
```

NOTE: cardInfo is a related command.

cardContactGet

Displays linecard contact information.

Syntax:

```
cardContactGet
```

Options:

None.

Sample Output:

```
-> cardContactGet
Card Contact        : John Doe
```

NOTE: cardInfo is a related command.

cardContactSet

Modifies the linecard contact information.

Syntax:

```
cardContactSet <string>
```

Options:

string

Contact text string.

Sample Output:

```
-> cardContactSet "Joe Smith"
Card Contact        : Joe Smith
```

NOTE: cardInfo is a related command.

cardNameGet

Displays the linecard name.

Syntax:

```
cardNameGet
```

Options:

None.

Sample Output:

```
-> cardNameGet
Card Name           : FVIC-6
```

NOTE: cardInfo is a related command.

cardNameSet

Modifies the linecard name.

Syntax:

```
cardNameSet <string>
```

Options:

string
Name text string.

Sample Output:

```
-> cardNameSet EVIC-6
Card Name           : EVIC-6
```

NOTE: cardInfo is a related command.

cardLocationGet

Displays the linecard location.

Syntax:

```
cardLocationGet
```

Options:

None.

Sample Output:

```
-> cardLocationGet
Card Location       : Main Chassis Unit, Slot 4
```

NOTE: cardInfo is a related command.

cardLocationSet

Modifies the linecard location information.

Syntax:

```
cardLocationSet <string>
```

Options:

string

Location text string.

Sample Output:

```
-> cardLocationSet "Lower Chassis, Slot 6"
Card Location       : Lower Chassis, Slot 6
```

NOTE: cardInfo is a related command.

Firmware

fwVersion

Displays the firmware versions for a unit.

Syntax:

```
fwVersion [slot]
```

Options:

slot

Slot number.

Sample Output:

```
-> fwVersion 3
Slot   3 Information ------------------
  Firmware Version: 4.0.0.0.32
  Firmware build:   4_0_0_0_32
  Firmware BSP:     VIO
  MBC Version:      None
  Bootrom Version:  4.0.0.0.28
```

bootQuery

Displays boot image version information.

Syntax:

```
bootQuery slot [-active | -alternate | -all]
```

Options:

-active
Displays the version of the active firmware image.

-alternate
Displays the version of the alternate firmware image.

-all
Displays the versions for the primary and alternate firmware images.

Sample Output:

```
-> bootQuery <SLOT NUMBER> -all
Primary firmware version: 4.0.0.0.32
Alternate firmware version: 4.0.0.0.28
Active firmware version: 4.0.0.0.32
```

bootSelect

Select the next boot image to be used.

Syntax:

```
bootSelect slot [-i index] [-alternate] [-version
version] [-noprompt]
```

Options:

slot
The slot number using the next boot image.

-i index
The index of the boot image to be used next.

-alternate
Chooses the alternate image to be used next.

-version version
Chooses a specific version to be the image to be used next.

-noprompt
Displays the current configuration only.

Sample Output:

```
-> bootSelect 103 -noprompt
Currently installed firmware versions
index : alias    : version
--------------------------
   1   : image1   : 4.0.0.0.28;
*# 2   : image2   : 4.0.0.0.32;

  * - indicates Default image (will run at next reboot)
  # - indicates Active image

Default boot image index = 2
```

Log

NOTE: For detailed information on commands for the Log group, refer to page 2-74.

Capture

capture

Display information for this device.

Syntax:

```
capture
```

Options:

None.

NOTE: This command is intended for support personnel.

captureFw

Display firmware information for this device.

Syntax:

```
captureFw
```

Options:

None.

NOTE: This command is intended for support personnel.

captureSm

Displays all available subnet manager information for module.

Syntax:

```
captureSm
```

Options:

None.

NOTE: This command is intended for support personnel.

captureLog

Display log information for this device.

Syntax:

```
captureLog
```

Options:

None.

NOTE: This command is intended for support personnel.

captureMisc

Display miscellaneous information for this device.

Syntax:

```
captureMisc
```

Options:

None.

NOTE: This command is intended for support personnel.

captureEthernet

Display pertinent Ethernet information for an EVIC.

Syntax:

```
captureEthernet
```

Options:

None.

NOTE: This command is intended for support personnel.

captureFibreChannel

Display pertinent Fibre Channel information for an FVIC.

Syntax:

```
captureFibreChannel
```

Options:

None.

NOTE: This command is intended for support personnel.

captureLinecard

Displays pertinent linecard information for module.

Syntax:

```
captureLinecard
```

Options:

None.

NOTE: This command is intended for support personnel.

captureShell

Display shell command information for this device.

Syntax:

```
captureShell
```

Options:

None.

NOTE: This command is intended for support personnel.

FibreChannel

fcPortSpeedGet

Retrieve the Fibre Channel port speed settings.

Syntax:

```
fcPortSpeedGet [Port Number]
```

Options:

PortNumber

Displays the current speed settings for a specific port.

Sample Output:

```
-> fcPortSpeedGet
                                                        Admin Link   Oper
Port Alias      PortWWN         NPortId     Topology    Speed Status Speed SFP
Status
Ext1 Ext-1  0x500066A1E0000133 0x010E00 Fabric         Auto  Up     4G    OK
Ext2 Ext-2  0x500066A2E0000133 0x000001 Private Loop   4G    Up     4G    OK
Ext3 Ext-3  0x500066A3E0000133 0x010A00 Fabric         Auto  Up     4G    OK
Ext4 Ext-4  0x500066A4E0000133 0x010D00 Fabric         Auto  Up     4G    OK
Ext5 Ext-5  0x500066A5E0000133 0x010B00 Fabric         Auto  Up     4G    OK
Ext6 Ext-6  0x500066A6E0000133 0x010C00 Fabric         Auto  Up     4G    OK
Ext7 Ext-7  0x500066A7E0000133 0x010B00 Fabric         Auto  Up     4G    OK
Ext8 Ext-8  0x500066A8E0000133 0x010C00 Fabric         Auto  Up     4G    OK
```

fcPortSpeedSet

Change the Fibre Channel port speed settings.

Syntax:

```
fcPortSpeedSet [port number] [speed]
```

Options:

port number

The Fibre Channel port number to set.

speed

New port speed in gigabytes/second (1,2, or 4). Choose 0 to auto-negotiate the highest speed.

Sample Output:

```
-> fcPortSpeedSet 1 1
Setting speed to 1
Updated the topology. Note that the board needs to be rebooted in order for the
change to take effect
                                                    Admin Link   Oper
Port Alias      PortWWN         NPortId     Topology    Speed Status Speed SFP
Status
Ext1 Ext-1  0x500066A1E0000133 0x010E00 Fabric         1G    Up     4G    OK

This Fibre Channel I/O device MUST BE REBOOTED to activate changes made with
fcPortSpeedSet
```

fcPortTopologyGet

Retreive the Fibre Channel port topology settings.

Syntax:

```
fcPortTopologyGet [port number]
```

Options:

port number

Displays the current topoloogy settings for a specified port.

Sample Output:

```
-> fcPortTopologyGet
                                                    Admin Link   Oper
Port Alias      PortWWN         NPortId     Topology    Speed Status Speed SFP
Status
Ext1 Ext-1  0x500066A1E0000133 0x010E00 Fabric         Auto  Up     4G    OK
Ext2 Ext-2  0x500066A2E0000133 0x000001 Private Loop   4G    Up     4G    OK
Ext3 Ext-3  0x500066A3E0000133 0x010A00 Fabric         Auto  Up     4G    OK
Ext4 Ext-4  0x500066A4E0000133 0x010D00 Fabric         Auto  Up     4G    OK
Ext5 Ext-5  0x500066A5E0000133 0x010B00 Fabric         Auto  Up     4G    OK
Ext6 Ext-6  0x500066A6E0000133 0x010C00 Fabric         Auto  Up     4G    OK
Ext5 Ext-7  0x500066A7E0000133 0x010B00 Fabric         Auto  Up     4G    OK
Ext6 Ext-8  0x500066A8E0000133 0x010C00 Fabric         Auto  Up     4G    OK
```

fcPortTopologySet

Change the Fibre Channel port topology settings.

Syntax:

```
fcPortTopologySet [PortNumber] [Topology]
```

Options:

PortNumber

The Fibre Channel port number to set.

Topology

New port topology:

0 = Private Loop

1 = Public Loop

2 = Point to Point

3 = Fabric

Sample Output:

```
-> fcPortTopologySet 5 1
Updated the topology. Note that the board needs to be rebooted in order for the
change to take effect
                                                    Admin Link   Oper
Port Alias      PortWWN         NPortId     Topology    Speed Status Speed SFP
Status
Ext5 Ext-5  0x500066A5E0000133 0x010B00 Public Loop    Auto  Up     4G    OK
```

fcPortAliasGet

Retrieves the Fibre Channel port alias settings.

Syntax:

```
fcPortAliasGet - Show Fibre Channel Port Alias (name)
```

Options:

PortNumber

Show the current alias (name) setting for a specified port. Executing this command without this option will display current alias settings for all Fibre Channel ports

Sample Output:

```
-> fcPortAliasGet


                                                    Admin Link   Oper
Port Alias      PortWWN         NPortId     Topology    Speed Status Speed SFP Status

   1 plc2a  0x500066A1DD000019 0x000000 Fabric         Auto  Down   ---   OK

   2 plc2b  0x500066A2DD000019 0x000000 Fabric         Auto  Down   ---   Not Installed

   3 plc2c  0x500066A3DD000019 0x000000 Fabric         Auto  Down   ---   Not Installed

   4 plc2d  0x500066A4DD000019 0x000000 Fabric         Auto  Down   ---   Not Installed

   5 plc2e  0x500066A5DD000019 0xF1F1F1 Private Loop   Auto  Down   ---   OK

   6 plc2f  0x500066A6DD000019 0x000000 Fabric         Auto  Down   ---   Not Installed

   7 plc2g  0x500066A7DD000019 0xF1F1F1 Fabric         Auto  Down   ---   Not Installed

   8 plc2h  0x500066A8DD000019 0xF1F1F1 Fabric         Auto  Down   ---   Not Installed
```

fcPortAliasSet

Changes the Fibre Channel port alias (name) setting.

Syntax:

```
fcPortAliasSet - Set Fibre Channel Port Alias (name)
```

Options:

PortNumber

The Fibre Channel port number to set.

Alias

New port alias (name).

Sample Output:

```
-> fcPortAliasSet 5 portX
```

NOTE: The port alias name is limited to six characters

fcDeviceDiscoveryStart

Initiates a discovery cycle for Fibre Channel devices.

Syntax:

```
fcDeviceDiscoveryStart
```

Options:

None.

Sample Output:

```
-> fcDeviceDiscoveryStart

Fibre Channel Device Discovery cycle started...
```

fcDiscoveredDevicesGet

Display Fibre Channel devices found during the last discovery cycle.

Syntax:

```
fcDiscoveredDevicesGet
```

Options:

Index

Optional index of the device from the discovered device list.

Sample Output:

```
-> fcDiscoveredDevicesGet
Ref#        NodeWWN             PortWWN       Port#  NPortId Name
1     0x500066A0DD000008 0x500066A4DD000008   3    0x010000 FVIC 00066a00dd000008
Port 4
2     0x500066A0DD000008 0x500066A4DD000008   5    0x010000 FVIC 00066a00dd000008
Port 4
3     0x500066A0DD000008 0x500066A4DD000008   6    0x010000 FVIC 00066a00dd000008
Port 4
4     0x500066A0DD000008 0x500066A4DD000008   4    0x010000 FVIC 00066a00dd000008
Port 4
```

fcConfiguredDevicesGet

Display Fibre Channel devices that have already been configured.

Syntax:

```
fcConfiguredDevicesGet
```

Options:

Index

Optional index of the device from the configured device list.

Sample Output:

```
-> fcConfiguredDevicesGet
Ref#        NodeWWN             PortWWN       Port#  NPortId COS ConnectStatus Name
1     0x20000004CF8C02E7 0x22000004CF8C02E7   1   0x0106d1  3  Connected     2e7
p1
2     0x50060E801042B920 0x50060E801042B920   1   0x0108ef  3  Connected
Hitachi P1 via p1
3     0x50060E801042B920 0x50060E801042B920   3   0x0108ef  3  Connected
hitachi p1 via p3
4     0x50060E801042B920 0x50060E801042B920   4   0x0108ef  3  Connected
hitachi p1  via p4
```

fcDiscoveredDeviceConfig

Configure Fibre Channel devices that have been discovered.

Syntax:

```
fcDiscoveredDeviceConfig [Device Index] [Name]
```

Options:

Device Name

The index number from the discovery list.

Name

The Fibre Channel target device name.

Sample Output:

```
-> fcDiscoveredDeviceConfig 27 "e31 p6"

Configuring discovered FC device 27 as (e31 p6)
```

fcConfiguredDeviceRem

Remove a Fibre Channel target device from the configured list.

Syntax:

```
fcConfiguredDeviceRem [Device Index]
```

Options:

Device Index

The index number from the configured Fibre Channel target device list.

Sample Output:

```
-> fcConfiguredDeviceRem 2
Removed FC Target Device 2.
```

fcConfiguredDeviceRename

Rename Fibre Channel devices that have already been configured

Syntax:

```
fcConfiguredDeviceRename <Device Index> <Name>
```

Options:

Device Index

Index number from the results of the configured device list.

Name

New name for the device.

Sample Output:

```
-> fcConfiguredDeviceRename 5 Renamed-device
Renaming Configured FC device 5 as (Renamed-device)
```

fcSrpInitiatorDiscoveryStart

Initiate a discovery cycle for SRP Initiators.

Syntax:

```
fcSrpInitiatorDiscoveryStart
```

Options:

None.

Sample Output:

```
-> fcSrpInitiatorDiscoveryStart

SRP Initiator Discovery cycle started...
```

fcSrpDiscoveredInitiatorsGet

Displays the SRP Initiators found during the last discovery cycle.

Syntax:

```
fcSrpDiscoveredInitiatorsGet
```

Options:

None.

Sample Output:

```
-> fcSrpDiscoveredInitiatorsGet
Ref#       Initiatorport Identifier
1      (0002C902003FFFFC:0000000000000001)
2      (0002C902003FFFFC:0000000000000002)
3      (0002C902003FFFFC:0000000000000003)
4      (0002C902003FFFFC:0000000000000004)
5      (00066A00A0006C20:0000000000013807)
```

fcSrpConfiguredInitiatorsGet

Display SRP Initiators that have already been configured.

Syntax:

```
fcSrpConfiguredInitiatorsGet
```

Options:

None.

Sample Output:

```
-> fcSrpConfiguredInitiatorsGet

Ref#       IB GUID      SRP Extension     ActiveConnections Name
1     (00066A00A0006801:0000000000009601)         0           st96
2     (00066A00A0006801:0000000000009602)         1           st9602
3     (00066A00A0006801:0000000000009603)         0           st9603
4     (00066A00A0006801:0000000000009604)         0           st9604
5     (00066A00A0006801:0000000000009605)         0           st9605
6     (00066A00A0006801:0000000000009606)         0           st9606
7     (00066A00A0007115:0000000000013601)         0           st136 01
8     (00066A00A0003F87:0000000000008203)         0           st82 03
11    (00066A00A0006CA8:0000000000010004)         0           st100 04
12    (00066A00A0006CA0:0000000000004205)         0           st42 05
13    (00066A00A0006C20:0000000000013806)         1           st138 06
14    (00066A00A00002D5:0000000000003901)         0           st3901
15    (00066A00A00002D5:0000000000000001)         0          st39, ext01
16    (00066A00A00002D5:0000000000000002)         0          st39, ext02
```

fcDiscoveredSrpInitiatorConfig

Configure a discovered SRP Initiator by specifying the index and name.

Syntax:

```
fcDiscoveredSrpInitiatorConfig - Configure a discovered
SRP Initiator
```

Options:

SRP Initiator Index

SRP Initiator Index.

Name

SRP Initiator Name.

Sample Output:

```
-> fcDiscoveredSrpInitiatorConfig
```

NOTE: The SRP Initiator Index is the number in the first column in the discovered initiators list. The discovered initiators list may be retrieved using the fcSrpDiscoveredInitiatorsGet command. Use double quotes around the name if it includes whitespace.

fcSrpInitiatorCreate

Create an new SRP Initiator by specifying the GUID, extension, and name.

Syntax:

```
fcSrpInitiatorCreate [Initiator GUID] [SRP Extension]
[Name]
```

Options:

Initiator GUID

The InfiniBand globally unique identifier (GUID) of the SRP Initiator.

SRP Extension

The SRP extension ID to use with the GUID.

Name

The SRP initiator name.

Sample Output:

```
-> fcSrpInitiatorCreate 0x00066A00A0006C20 0x013807 "st138 08"

InitiatorGUID[00066A00A0006C20] SrpExtension[0000000000013808] st138 08

Ref#       IB GUID      SRP Extension     ActiveConnections Name
21    (00066A00A0006C20:0000000000013808)        0          st138 08
```

NOTE: Both the Initiator GUID and SRP Extension values must be specified in hexadecimal format using leading 0x or 0X prefixes. If a wildcard GUID is desired a single zero is permitted. Use double quotes around the name if it includes whitespace..

fcSrpInitiatorRem

Remove a SRP Initiator from the configured list.

Syntax:

```
fcSrpInitiatorRem [Initiator Index]
```

Options:

Initiator Index

The initiator number that is displayed in the configured initiator list.

Sample Output:

```
-> fcSrpInitiatorRem 21
```

fcSrpMapShowConnections

Retrieve SRP initiator connection information.

Syntax:

```
fcSrpMapShowConnections <Initiator Index>
```

Options:

Initiator Index

An initiator number from the configured initiator list. Executing this command without this option will display all initiator connections.

Sample Output:

```
-> fcSrpMapShowConnections
```

fcSrpMapConfigGet

Retrieve SRP map configuration information.

Syntax:

```
fcSrpMapConfigGet [Initiator Index]
```

Options:

Initiator Index

The intiator index that is associated with the configured SRP map.

Sample Output:

```
-> fcSrpMapConfigGet

Initiator 1, (st96) using GUID: 00066A00A0006801 SRP Ext:0000000000009601
   IOC 1 SRP Map, (p1 hitachi) is a DIRECT type. It contains 1 LUN map(s).
      All Host LUNs are passed directly to the Target (Hitachi P1 via p1)

Initiator 2, (st9602) using GUID: 00066A00A0006801 SRP Ext:0000000000009602
   IOC 1 SRP Map, (st96 hit over p2) is a DIRECT type. It contains 1 LUN map(s).
      All Host LUNs are passed directly to the Target (htachi p1 via p2)

Initiator 17, (uc blade ext1) using GUID: 0005AD000002BA4E SRP
Ext:0000000000000001

Initiator 18, (uc blade ext2) using GUID: 0005AD000002BA4E SRP
Ext:0000000000000002

Initiator 20, (uc blade ext4) using GUID: 0005AD000002BA4E SRP
Ext:0000000000000004
   IOC 1 SRP Map, (maaaap) is a EXPLICIT type. It contains 3 LUN map(s).
      Host LUN 0 is mapped to Target LUN 0 on (2e7 p1)
      Host LUN 1 is mapped to Target LUN 0 on (d94 p3)
      Host LUN 2 is mapped to Target LUN 0 on (d88 p4)
   IOC 2 SRP Map, (paaaam) is a DIRECT type. It contains 1 LUN map(s).
      All Host LUNs are passed directly to the Target (b57 p4)


[Initiators 1 & 2 have one direct map. Initiators 17 & 18 have no maps. Initiator
20 has two maps:
 explicit on IOC1 and direct on IOC2.]
```

fcSrpMapConfigCreate

Create and configure a new SRP map association.

Syntax:

```
fcSrpMapConfigCreate [Initiator Index] [IOC] [FC Target
Device Index] [MapType] [MapName]
```

Options:

Initiator Index

The index number from the Configured SRP Initiators list.

IOC

The IOC number (either 1 or 2).

FC Target Device Index

The index number from the Configured FC Target Device list.

MapType

The type of SRP Map (D=Direct or E=Explicit).

MapName

The SRP Map name.

Sample Output:

```
-> fcSrpMapConfigCreate 17 1 14 D neptune

Creating Direct Map...
Activating Map...
Initiator 17, (uc blade ext1) using GUID: 0005AD000002BA4E SRP
Ext:0000000000000001
   IOC 1 SRP Map, (neptune) is a DIRECT type. It contains 1 LUN map(s).
      All Host LUNs are passed directly to the Target (b57 p4)
```

NOTE: Use double quotes around the Map Name if it includes whitespace. The Target Device Index parameter is required for creation of DIRECT map types. Do not provide the Target Device Index parameter when creating EXPLICIT maps.

fcSrpMapConfigStatus

Change the status of a map.

Syntax:

```
fcSrpMapConfigStatus [SRP Initiator Index] [IOC]
[Status]
```

Options:

SRP Initiator Index

The index number from the Configured SRP Initiators list.

IOC

The IOC number (either 1 or 2).

Status

The type of SRP Map (A=Active or I=Inactive)

Sample Output:

```
-> fcSrpMapConfigStatus 18 1 I

De-activating Map...
Initiator 18, (uc blade ext2) using GUID: 0005AD000002BA4E SRP
Ext:0000000000000002
   IOC 1 SRP Map, (jupiter) is a EXPLICIT type. It contains 2 LUN map(s). STATUS:
INACTIVE
      Host LUN 0 is mapped to Target LUN 0 on (b57 p4)
      Host LUN 1 is mapped to Target LUN 0 on (PerfTarget)
```

NOTE: SRP Maps that have active host connections may not be deactivated. Deactivate the host connections before attempting.

fcSrpMapConfigRowAdd

Add a LUN association to a SRP map.

Syntax:

```
fcSrpMapConfigRowAdd [Initiator Index] [IOC] [FC Target
Device Index] [HostLUN] [TargetLUN]
```

Options:

Initiator Index

The index number from the Configured SRP Initiators list.

IOC

The IOC number (either 1 or 2).

FC Target Device Index

The index number from the Configured FC Target Device list.

HostLUN

The host Logical Unit Number.

TargetLUN

The target Logical Unit Number.

Sample Output:

```
-> fcSrpMapConfigRowAdd 18 1 14 0 0

Initiator 18, (uc blade ext2) using GUID: 0005AD000002BA4E SRP
Ext:0000000000000002
   IOC 1 SRP Map, (jupiter) is a EXPLICIT type. It contains 1 LUN map(s). STATUS:
INACTIVE
      Host LUN 0 is mapped to Target LUN 0 on (b57 p4)
```

NOTE: The specified SRP map must be explicit type and already exist

fcSrpMapConfigRowRem

Remove a LUN association from SRP map.

Syntax:

```
fcSrpMapConfigRowRem [Initiator Index] [IOC] [FC Target
Device Index] [HostLUN] [TargetLUN]
```

Options:

Initiator Index

The index number from the Configured SRP Initiators list.

IOC

The IOC number (either 1 or 2).

FC Target Device Index

The index number from the Configured FC Target Device list.

HostLUN

The host Logical Unit Number.

TargetLUN

The target Logical Unit Number.

Sample Output:

```
-> fcSrpMapConfigRowRem 18 1 15 1 0

Initiator 18, (uc blade ext2) using GUID: 0005AD000002BA4E SRP
Ext:0000000000000002
   IOC 1 SRP Map, (jupiter) is a EXPLICIT type. It contains 1 LUN map(s). STATUS:
INACTIVE
      Host LUN 0 is mapped to Target LUN 0 on (b57 p4)
```

NOTE: The specified SRP map must be explicit type and already exist

fcSrpMapConfigRem

Delete a SRP map from an existing map configuration.

Syntax:

```
fcSrpMapConfigRem [InitiatorIndex] [IOC]
```

Options:

Initiator Index

The index number from the Configured SRP Initiators list.

IOC

The IOC number (either 1 or 2).

Sample Output:

```
-> fcSrpMapConfigRem 18 1

Initiator 18, (uc blade ext2) using GUID: 0005AD000002BA4E SRP
Ext:0000000000000002
```

NOTE: SRP Maps that have active host connections may not be deleted. Disconnect the host connections before attempting removal.

fcSrpMapConfigType

Allows the user to change the SRP map type (Direct or Explicit).

Syntax:

```
fcSrpMapConfigType <SRP Initiator Index> <IOC> <MapType>
```

Options:

SRP Initiator Index

The index number from the Configured SRP Initiators list.

IOC

The IOC number (either 1 or 2).

MapType

Type of SRP Map (D=Direct or E=Explicit).

Host LUN

Optional: if not used, the Host LUN will be set to 0.

Target LUN

Optional: if not used, the Target LUN will set to 0.

Sample Output:

```
-> fcSrpMapConfigType Initiator1 1 D
```

NOTE: If there is only 1 LUN in an Explicit map the user should change from an Explicit map to a Direct Map

fcSrpMapConfigName

Allows the user to change the SRP map name.

Syntax:

```
fcSrpMapConfigName <SRP Initiator Index> <IOC> <MapName>
```

Options:

SRP Initiator Index

The index number from the Configured SRP Initiators list.

IOC

The IOC number (either 1 or 2).

MapName

SRP map name.

Sample Output:

```
-> fcSrpMapConfigName Initiator2 1 Map3
```

NOTE: Use double quotes around the Map Name if it includes whitespace.

fcTargetDeviceStatsGet

Display the Fibre Channel Target Device statistics.

Syntax:

```
fcTargetDeviceStatsGet
```

Options:

<Device>

Optional Device Index.

Sample Output:

```
-> fcTargetDeviceStatsGet
Ref#        NodeWWN             PortWWN      Port#  NPortId COS
ConnectStatus Name
Device 1: 2e7 p1
  NodeWWN=20000004CF8C02E7
  PortWWN=22000004CF8C02E7
  PortIndex=1 ConnectStatus=Connected
  NPortId=0x0106d1 ClassOfService=3
  Size: in=2048 out=2048
  TotalRequests=0
  SucceededRequests=0
  FailedRequests=0
  OutstandingRequests=0
  TotalDataIn=0
  TotalDataOut=0

Device 2: Hitachi P1 via p1
  NodeWWN=50060E801042B920
  PortWWN=50060E801042B920
  PortIndex=1 ConnectStatus=Connected
  NPortId=0x0108ef ClassOfService=3
  Size: in=2048 out=2048
  TotalRequests=0
  SucceededRequests=0
  FailedRequests=0
  OutstandingRequests=0
  TotalDataIn=0
  TotalDataOut=0
```

NOTE: Invoking this command without a device index displays the statistics for all devices.

fcSrpInitiatorStatsGet

Display the SRP Initiator Statistics.

Syntax:

```
fcSrpInitiatorStatsGet
```

Options:

```
<Initiator>
```
Optional Initiator Index.

Sample Output:

```
-> fcSrpInitiatorStatsGet

Initiator 1, (st96) using GUID: 00066A00A0006801 SRP
Ext:0000000000009601
  ActiveConnections   = 0
  TotalRequests       = 0
  SucceededRequests   = 0
  FailedRequests      = 0
  OutstandingRequests = 0
  TotalDataIn         = 0
  TotalDataOut        = 0

Initiator 2, (st9602) using GUID: 00066A00A0006801 SRP
Ext:0000000000009602
  ActiveConnections   = 1
  TotalRequests       = 23553
  SucceededRequests   = 23545
  FailedRequests      = 8
  OutstandingRequests = 0
  TotalDataIn         = 0
  TotalDataOut        = 0
```

NOTE: Invoking this command without an initiator index displays statistics for all initiators.

fcPortStatsGet

Display Fibre Channel Port Statistics.

Syntax:

```
fcPortStatsGet [PortNumber]
```

Options:

PortNumber

The Fibre Channel port number for the device.

Sample Output:

```
-> fcPortStatsGet
Port Ext1: VIO 00066a00e0000133 Port 1
  Alias        : Ext-1
  NodeWWN      : 500066A0E0000133
  PortWWN      : 500066A1E0000133
  NPortId      : 0x010e00
  Topology     : Fabric
  Admin Speed : Auto
  LinkStatus   : Up
  Oper Speed   : 4G
  SFP State    : OK
  LossOfSignalCount     : 73
  LossOfSyncCount       : 2
  InvalidTransWordCount : 510
  LinkFailureCount      : 2
  PrimitiveSeqCount     : 0
```

NOTE: Invoking this command without a portNumber displays statistics for all Fibre Channel ports.

fcTrapStatusGet

Retrieve the list and status of all Fibre Channle traps.

Syntax:

```
fcTrapStatusGet
```

Options:

None.

Sample Output:

```
-> fcTrapStatusGet
Valid Traps:
 1) Interconnect Link Up      : enabled
 2) Interconnect Link Down    : enabled
 3) Target Device Link Up     : enabled
 4) Target Device Link Down   : enabled
```

fcTrapEnable

Enable specified Fibre Channel traps.

Syntax:

```
fcTrapEnable [Trap Index]
```

Options:

Trap Index
The index number of the trap to enable.

Sample Output:

```
-> fcTrapEnable 1
Trap 'Interconnect Link Up' enabled
```

NOTE: The Trap Index may be retrieved using fcTrapStatusGet.

fcTrapDisable

Disable a specified Fibre Channel trap.

Syntax:

```
fcTrapDisable [Trap Index]
```

Options:

Trap Index
The index number of the trap to disable.

Sample Output:

```
-> fcTrapDisable 1
Trap 'Interconnect Link Up' disabled
```

NOTE: The Trap Index may be retrieved using fcTrapStatusGet.

fcTrapGenerate

Generate a specified Fibre Channel trap.

Syntax:

```
fcTrapGenerate
```

Options:

Trap Index
The index number of the trap to enable.

Sample Output:

```
-> fcTrapGenerate 1
Generated trap for Interconnect Link Up
```

NOTE: The Trap Index may be retrieved using fcTrapStatusGet.

fcVirtPortShow

Displays information about the virtual ports in the pool

Syntax:

```
fcVirtPortShow <Index>
```

Options:

Index
Optional Virtual Port Index.

Sample Output:

```
-> fcVirtPortShow
WWN Key: T4WD5G-S6A5U2-QA28A7-32QKS8-ZS8AW7-G

Index |         WWN         | Alias |  PhyPort   |  Status  | Name
------------------------------------------------------------------------
--
 0001 | 0x500066A0E2000199 | Ext-1 |     1      | Active   | Test-0001
 0002 | 0x500066A0E2200199 |       | Unassigned | Inactive | WWN-0002
 0003 | 0x500066A0E2400199 |       | Unassigned | Inactive | WWN-0003
 0004 | 0x500066A0E2600199 |       | Unassigned | Inactive | WWN-0004
 0005 | 0x500066A0E2800199 |       | Unassigned | Inactive | WWN-0005
 0006 | 0x500066A0E2A00199 |       | Unassigned | Inactive | WWN-0006
 0007 | 0x500066A0E2C00199 |       | Unassigned | Inactive | WWN-0007
…
…
…
0128 | 0x500066AFE2E00199 |       | Unassigned | Inactive | WWN-0128
```

fcVirtPortNameSet

Set the text name field of the virtual port

Syntax:

```
fcVirtPortNameSet <Index> <Name>
```

Options:

Index

The index from the Virtual Port list.

Name

The text name for the Virtual Port.

Sample Output:

```
-> fcVirtPortNameSet 1, Test-0001

Index |         WWN         | Alias |  PhyPort   |  Status  | Name
-------------------------------------------------------------------------
 0001 | 0x500066A0E2000199 |       | Unassigned | Inactive | Test-0001

fcVirtPortPhyPortSet:
Configure Virtual Port index 1 to use physical port 1

-> fcVirtPortPhyPortSet 1, 1

Index |         WWN         | Alias |  PhyPort   |  Status  | Name
-------------------------------------------------------------------------
 0001 | 0x500066A0E2000199 | Ext-1 |     1      | Inactive | Test-0001
```

NOTE: The text name field is limited to 32 characters.

fcVirtPortPhyPortSet

Assign a physical port to a virtual port

Syntax:

```
fcVirtPortPhyPortSet <Index> <Port Number>
```

Options:

Index

The index from the Virtual Port list

Port Number

The Physical Port to be assigned

Sample Output:

```
-> fcVirtPortPhyPortSet 1 2

Index |   VirtPort WWN    |  PhyPort   |  Status  | VirtPort Name
-------------------------------------------------------------------------
 0001 | 500066A0E2000199 |     2      |  Inactive  | WWN-0001
```

NOTE: The virtual port must be inactive for the assignment to be successful. Use port zero to unassign a virtual port.

fcVirtPortStatusSet

Activates or deactivates the virtual port.

Syntax:

```
fcVirtPortStatusSet <Index> <Status>
```

Options:

Index

The index from the Virtual Port list.

Status

A=Active, I=Inactive

Sample Output:

```
-> fcVirtPortStatusSet 1, 1

Index |         WWN         | Alias |  PhyPort   |  Status  | Name
-------------------------------------------------------------------------
 0001 | 0x500066A0E2000199 | Ext-1 |     1      | Active   | Test-0001
```

NOTE: The virtual port must have an assigned physical port in order to be activated.

fcVirtPortPoolCreate

Create the Virtual Port Pool from the feature key.

Syntax:

```
fcVirtPortPoolCreate <KeyString>
```

Options:

KeyString

Encoded feature key to enable the Virtual Port Pool.

Sample Output:

```
-> fcVirtPortPoolCreate <license key>
Virtual Port Pool key accepted.

  Virtual Port Pool Key: <license key>

Index |   VirtPort WWN   |  PhyPort   |  Status  | VirtPort Name
-------------------------------------------------------------------
 0001 | 500066A0E2000199 | Unassigned | Inactive | WWN-0001
 0002 | 500066A0E2200199 | Unassigned | Inactive | WWN-0002
 0003 | 500066A0E2400199 | Unassigned | Inactive | WWN-0003
.
.
.
 0126 | 500066AFE2A00199 | Unassigned | Inactive | WWN-0126
 0127 | 500066AFE2C00199 | Unassigned | Inactive | WWN-0127
 0128 | 500066AFE2E00199 | Unassigned | Inactive | WWN-0128
```

NOTE: To enable this feature, acquire the feature key from QLogic.

fcVirtPortPoolRem

Remove all Virtual Port entries.

Syntax:

```
fcVirtPortPoolRem <KeyString>
```

Options:

KeyString

The encoded feature key that was previously used to enable the Virtual Port Pool.

Sample Output:

```
->fcVirtPortPoolRem <KeyString>
The Virtual Port Pool was removed.
```

NOTE: WARNING: This operation will cause all active virtual ports to be disconnected and all persistent virtual port information will be deleted. In order to restore the virtual port capability, the feature key will need to be re-entered.

Ethernet

ethMtuGet

Read the current and saved maximum transfer unit (MTU) sizes.

Syntax:

```
ethMtuGet
```

Options:

None.

Sample Output:

```
ethMtuGet
Ethernet MTU size:
  Current : 9500
  Saved   : 9500
```

NOTE: Prints the current and saved Ethernet MTU sizes for the system. The current MTU is the size currently in use, while the saved MTU is the size that will be used following a reboot.

ethMtuSet

Set the MTU size to be used after the next reboot

Syntax:

```
ethMtuSet [mtuSize]
```

Options:

mtuSize

Size (in bytes) of the new MTU.

Sample Output:

```
ethMtuSet 1500
Successfully set MTU size to 1500.  A reboot is required for the change
to take effect.
```

NOTE: Sets the Ethernet MTU size to a specified value. Once set, the MTU is saved will take effect following a reboot.

ethViPortsGet

Displays the VIPORT counts.

Syntax:

```
ethViPortsGet
```

Options:

None.

Sample Output:

```
-> ethViPortsGet
ViPorts:
  Current :  250
  Saved   :  250
  Maximum :  400
  MAC Addr:  250
```

NOTE: Prints the current and saved VIPORT counts for the system. The current VIPORT count is the number of active host connections. The saved VIPORT count is the number that can be used following a reboot.

ethViPortsSet

Modifies the VIPORT count (takes effect after the next reboot).

Syntax:

```
ethViPortsSet
```

Options:

viPorts

Number of supported host connections.

Sample Output:

```
-> ethViPortsSet 25
Successfully set VIPORT count to 25.  A reboot is required for the
change to take effect.
```

NOTE: Sets the VIPORT count to the specified value. This is the saved VIPORT count and will not take effect until the next reboot.

ethExtPauseGet

Read whether PAUSE is enabled or disabled

Syntax:

```
ethExtPauseGet <port>
```

Options:

port
The external Ethernet port(s) to display the PAUSE state.

Sample Output:

```
ethExtPauseGet "1 2"
External Ethernet port 1 [Ext-1] PAUSE: enabled
External Ethernet port 2 [Ext-2] PAUSE: enabled
```

NOTE: Displays the PAUSE state for an external Ethernet port. The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

ethExtPriorityGet

Read default user priority for the specified ethernet port.

Syntax:

```
ethExtPriorityGet [portNumber]
```

Options:

portNumber
The Ethernet port for which to get default user priority.

Sample Output:

```
ethExtPriorityGet 1
External Ethernet port 1 default user priority: 2
```

NOTE: Displays the current default user priority for an external Ethernet port. The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

ethExtPrioritySet

Set default user priority for the specified ethernet port.

Syntax:

```
ethExtPrioritySet [port] [priority]
```

Options:

port

The Ethernet port(s) to set a default user priority.

priority

A new default user priority.

Sample Output:

```
ethExtPrioritySet 1 3
External Ethernet port 1: default user priority changed to 3
```

NOTE: Sets the default user priority for an external Ethernet port. Valid settings are in the range [0,7]. The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes

ethVirtPriorityGet

Read default user priority for the specified virtual port.

Syntax:

```
ethVirtPriorityGet [port]
```

Options:

port

The virtual port used to get the default user priority.

Sample Output:

```
ethVirtPriorityGet 1
Virtual Ethernet port 1 default user priority: 2
```

NOTE: Displays the current default user priority for a specified virtual port. The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

ethVirtPrioritySet

Modifies the default user priority for the specified virtual port.

Syntax:

```
ethVirtPrioritySet [port] [priority]
```

Options:

port

The virtual port(s) used to set the default user priority.

priority

A new default user priority.

Sample Output:

```
ethVirtPrioritySet 1 7
Virtual Ethernet port 1: default user priority changed to 7
```

NOTE: Sets the default user priority for a specified virtual port. Value settings are in the range [0,7]. The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

ethExtStatsGet

Displays statistics for a specified Ethernet port(s).

Syntax:

```
ethExtStatsGet [port]
```

Options:

port

The Ethernet port(s) from which the user chooses to have statistics displayed.

Sample Output:

```
ethExtStatsGet 1
Current stats for External Ethernet port 1 [Ext-1]:

  Drop Events            : 0x00000000 : 0
  Octets                 : 0x00019a51 : 105041
  Packets                : 0x0000048b : 1163
  Broadcast Pkts         : 0x00000054 : 84
  Multicast Pkts         : 0x00000437 : 1079
  CRC Align Errors       : 0x00000000 : 0
  Undersize Pkts         : 0x00000000 : 0
  Oversize Pkts          : 0x00000000 : 0
  Fragments              : 0x00000000 : 0
  Jabbers                : 0x00000000 : 0
  Collisions             : 0x00000000 : 0
  64 Octets              : 0x00000323 : 803
  65-127 Octets          : 0x00000126 : 294
  128-255 Octets         : 0x00000003 : 3
  256-511 Octets         : 0x0000003f : 63
  512-1023 Octets        : 0x00000000 : 0
  1024-1518 Octets       : 0x00000000 : 0

Alignment Errors             : 0x00000000 : 0
  FCS Errors                   : 0x00000000 : 0
  Single Collision Frames      : 0x00000000 : 0
  Multiple Collision Frames    : 0x00000000 : 0
  SQE Test Errors              : 0x00000000 : 0
  Deferred Transmissions       : 0x00000000 : 0
  Late Collisions              : 0x00000000 : 0
  ExcessiveCollisions          : 0x00000000 : 0
  Internal Mac Transmit Errors : 0x00000000 : 0
  Carrier Sense Errors         : 0x00000000 : 0
  Frame Too Longs              : 0x00000000 : 0
  Internal Mac Receive Errors  : 0x00000000 : 0
  Symbol Errors                : 0x00000000 : 0

  Out Octets             : 0x000014c0 : 5312
  Out Unicast Pkts       : 0x00000000 : 0
  Out Multicast Pkts     : 0x00000053 : 83
  Out Broadcast Pkts     : 0x00000000 : 0
  Out Discard            : 0x00000000 : 0
  Out Errors             : 0x00000000 : 0
```

NOTE: Prints various statistics for a specified external Ethernet port. The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

ethVirtStatsGet

Print statistics for a specified virtual port(s).

Syntax:

```
ethVirtStatsGet [port]
```

Options:

port

The virtual port(s) from which to have the statistics displayed.

Sample Output:

```
ethVirtStatsGet 1

Current stats for virtual port 1:
 Incoming
  Broadcast Pkts          : 0x00000066 : 102
  Multicast Pkts          : 0x00000017 : 23
  Unicast Pkts            : 0x00000000 : 0
  Undersize+Oversize      : 0x00000000 : 0

 Outgoing
  Broadcast Pkts          : 0x00000000 : 0
  Multicast Pkts          : 0x00000000 : 0
  Unicast Pkts            : 0x00000000 : 0

 General
  Directed Bytes Rcv      : 0x00000000 : 0
  Multicast Bytes Rcv     : 0x00000564 : 1380
  Broadcast Bytes Rcv     : 0x0000787a : 30842
  Directed Bytes Xmt      : 0x00000000 : 0
  Multicast Bytes Xmt     : 0x00000000 : 0
  Broadcast Bytes Xmt     : 0x00000000 : 0
```

NOTE: Prints various statistics for a specified virtual port. The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

vlanDot1qPvidGet

View/Set VLAN Dot1qPvid for a port.

Syntax:

```
vlanDot1qPvidGet <target> <instance>
```

Options:

target
eth|vi|ioc
instance(s)
port number/IOC GUID

NOTE: Displays the default VLAN Dot1qPvid for a specified external Ethernet port, Virtual Ethernet port, or IOC. The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

vlanDot1qPvidSet

View/Set VLAN Dot1qPvid for a port.

Syntax:

```
vlanDot1qPvidSet <target> <instance(s)> <pvid>
```

Options:

target
eth|vi|ioc
instance(s)
port number/IOC GUID
pvid
VLAN tag

NOTE: Changes the default VLAN Dot1qPvid for a specified external Ethernet port, Virtual Ethernet port, or IOC to the new value indicated by the pvid parameter. The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

vlanMapTaggedGet

Show VLAN PVID tagged map.

Syntax:

```
vlanMapTaggedGet <target> <instance>
```

Options:

target

eth|vi

instance(s)

port number

Sample Output:

```
-> vlanMapTaggedGet vi 56

vlanMapTaggedGet(0):
vi(2)   flag=0
-------------------------
Port=56  Total PVID=1
11

-----------------------------
-> vlanMapTaggedGet eth1
usage: vlanMapTaggedGet <target eth|vi> <instance(s)>
instance => Ethernet or Virtual port number
For multiple instances, enclose with ". E.g. "1 2 3"
-> vlanMapTaggedGet eth 1

vlanMapTaggedGet(0):
eth(1)  flag=0
-------------------------
Port=1 [Ext-1] Total PVID=2
11 36
-----------------------------
```

NOTE: Displays the VLAN PVID tagged map for Ethernet port(s). Related commands: vlanDot1qPvid, vlanMapMember.

vlanMapTaggedSet

Update VLAN PVID tagged map.

Syntax:

```
vlanMapTaggedSet <target> <instance(s)> <pvid(s)> <flag
0-remove|1-add>
```

Options:

target

eth|vi

instance(s)

port number

pvid(s)

VLAN tag(s)

flag

0-remove|1-add

NOTE: Update the VLAN PVID tagged map for Ethernet port(s). Related commands: vlanDot1qPvid, vlanMapMember.

vlanMapMemberGet

Show VLAN PVID member map.

Syntax:

```
vlanMapMemberGet <target> <instance(s)>
```

Options:

target

eth|vi

instance(s)

port number

Sample Output:

```
-> vlanMapMemberGet vi 56

vlanMapMemberGet(0):
vi(2) flag=0
-------------------------
Port=56  Total PVID=1
11
-----------------------------
-> vlanMapMemberGet eth 1

vlanMapMemberGet(0):
eth(1) flag=0
-------------------------
Port=1 [Ext-1]  Total PVID=2
11 36
```

NOTE: Displays the VLAN PVID member map for Ethernet or vitrual port(s). Related commands: vlanDot1qPvid, vlanMapMember.

vlanMapMemberSet

Update VLAN PVID member map.

Syntax:

```
vlanMapMemberSet <target> <instance(s)> <pvid(s)> <flag
0-remove|1-add>
```

Options:

target

eth|vi

instance(s)

port number

pvid(s)

VLAN tag(s)

flag

0-remove|1-add

Sample Output:

```
-> vlanMapMemberSet eth 1 11 0

vlanMapMemberSet(1):
eth(1) flag=0
-------------------------
Port=1 [Ext-1]  Total PVID=1
36
```

NOTE: Updates the VLAN PVID member map for Ethernet or virtual port(s). Related commands: vlanDot1qPvid, vlanMapMember.

vlanInfo

Display an overview of the VLAN configuration..

Syntax:

```
vlanInfo target
```

Options:

target
Determine whether to show External or Virtual Ethernet port configuration.

NOTE: Displays an overview of the VLAN configuration for all ports of the specified type. The target parameter may be eth, vi, or all for external Ethernet ports, virtual Ethernet ports, or all ports,respectively.

vlanMapInfo

Display VLAN mapping information.

Syntax:

```
vlanMapInfo target
```

Options:

target
Determine whether to show External or Virtual Ethernet port mapping information.

NOTE: Displays VLAN mapping information for the specified port type. The target parameter must be either eth (Ethernet), vi (virtual Ethernet) or all.

vlanEgressRuleGet

View/Set VLAN Egress rule(s).

Syntax:

```
vlanEgressRuleGet <target> <instance(s)>
```

Options:

target

eth|vi|ioc

instance(s)

port number(s)/IOC GUID

Sample Output:

```
-> vlanEgressRuleGet vi 56

vlanEgressRuleGet(0): Flag=0(1-enable|0-disable)
vi(2)   Result(>1 => error)
-------------------------
vi:56   1
-> vlanEgressRuleGet eth 1

vlanEgressRuleGet(0): Flag=0(1-enable|0-disable)
eth(1)  Result(>1 => error)
-------------------------
eth:1 [Ext-1]   1
```

NOTE: Displays the VLAN egress rule(s) for Ethernet/host port(s). Related commands: vlanIngressRule, vlanIngressFrame, vlanDot1qPvid.

vlanEgressRuleSet

View/Set VLAN Egress rule(s).

Syntax:

```
vlanEgressRuleSet <target> <instance(s)> <flag
1-enable|0-disable>
```

Options:

target

eth|vi|ioc

instance(s)

port number(s)/IOC GUID

flag

1-enable|0-disable

NOTE: Displays the VLAN egress rule(s) for Ethernet/host port(s). Related commands: vlanIngressRule, vlanIngressFrame, vlanDot1qPvid.

vlanIngressRuleGet

View/Set VLAN Ingress rule(s).

Syntax:

```
vlanIngressRuleGet <target> <instance(s)>
```

Options:

target

eth|vi|ioc

instance(s)

port number(s)/IOC GUID

Sample Output:

```
-> vlanIngressRuleGet vi 56

vlanIngressRuleGet(0):
vi(2)   flag[1-enable|2-disable]
-------------------------
vi:56   1
-> vlanIngressRuleGet eth 1

vlanIngressRuleGet(0):
eth(1)  flag[1-enable|2-disable]
-------------------------
eth:1 [Ext-1]   2
```

NOTE: Displays the VLAN Ingress rule flag for port(s) (Ethernet port/virtual port/IOC GUID). Related commands: vlanEgressRule, vlanIngressFrame, vlanDot1qPvid.

vlanIngressRuleSet

View/Set VLAN Ingress rule(s).

Syntax:

```
vlanIngressRuleSet <target> <instance(s)> <flag
1-enable|2-disable>
```

Options:

target

eth|vi|ioc

instance(s)

port number(s)/IOC GUID

flag

1-enable|0-disable

NOTE: Updates the VLAN Ingress rule flag for port(s) (Ethernet port/virtual port /IOC GUID). Related commands: vlanEgressRule, vlanIngressFrame, vlanDot1qPvid.

vlanIngressFrameGet

View/Set VLAN Ingress Frame type.

Syntax:

```
vlanIngressFrameGet <target> <instance(s)>
```

Options:

target
eth|vi|ioc

instance(s)
port number(s)/IOC GUID

NOTE: Displays the VLAN Ingress acceptable frame type for port(s)(Ethernet port/ViPort/IOC GUID). Related commands: vlanEgressRule, vlanIngressRule, vlanDot1qPvid.

vlanIngressFrameSet

View/Set VLAN Ingress Frame type.

Syntax:

```
vlanIngressFrameSet <target> <instance(s)> <flag 1-admit
all|2-admit VLAN tagged>
```

Options:

target
eth|vi|ioc

instance(s)
port number(s)/IOC GUID

flag
1-admit all|2-admit VLAN tagged

NOTE: Updates the VLAN Ingress acceptable frame type for port(s)(Ethernet port/Vi Port/IOC GUID). Related commands: vlanEgressRule, vlanIngressRule, vlanDot1qPvid.

ethMaxPacketAgeGet

Show the maximum packet age for a LAN switch.

Syntax:

```
ethMaxPacketAgeSet <value>
```

NOTE: Displays the maximum packet age (in milliseconds) for all LAN switches.

ethMaxPacketAgeSet

View/Set Ethernet maximum packet age.

Syntax:

```
ethMaxPacketAgeSet <value>
```

Options:

```
value
  in millisecond
```

Sample Output:

```
ethMaxPacketAgeSet 54999

ethMaxPacketAgeSet(1):
------------------------
max. packet age: System=54999 ms
```

NOTE: Sets the maximum packet age (in milliseconds) for all LAN switches.

vlanGet

View VLAN configuration.

Syntax:

```
vlanGet <type> <host> <ioc> [<instance(s)>]
```

Options:

type

name|guid

host

host name/guid

ioc

IOC number

instance

host interface instance(s) (optional)

Sample Output:

```
vlanGet name st130 1 0

vlanGet: action=0 target=4 defVlan=1 admitAll=1 IOC=1 PVID=1
hostInst=0




[Vi=81] [HostGUID=0x0002c90200216e18] [IOC=1] [Inst=0] [Name=st130]
---

ViPort: DefPVID=11 EgressFilter=yes IngressFilter=yes
IngressFrame=tagged

VLAN Map Membership:11

VLAN Map allowed tagging:--Empty--

ExtPort[1]: DefPVID=11 EgressFilter=yes IngressFilter=yes
IngressFrame=all

VLAN Map Membership:11

VLAN Map allowed tagging:11
```

NOTE: View the VLAN configuration to a host and IOC pair. Related commands: vlanSet, vlanRem.

vlanSet

Set the VLAN configuration.

Syntax:

```
vlanSet <type> <host> <ioc> <vlan> <pvid> [<admit>]
[<instance(s)>]
```

Options:

type
name|guid

host
host name/guid

ioc
IOC number

vlan type
default|host

pvid
default PVID

admit
1=tagged | 0=all (optional)

instance
host interface instance (optional)

Sample Output:

```
vlanSet name st130 1 13 13 1 0

vlanSet: action=1 target=4 defVlan=1 admitAll=0 IOC=1 PVID=13
hostInst=0




[Vi=81] [HostGUID=0x0002c90200216e18] [IOC=1] [Inst=0] [Name=st130]
---

ViPort: DefPVID=13 EgressFilter=yes IngressFilter=yes
IngressFrame=tagged

VLAN Map Membership:11

VLAN Map allowed tagging:--Empty--

ExtPort[1]: DefPVID=11 EgressFilter=yes IngressFilter=yes
IngressFrame=tagged

VLAN Map Membership:11,13

VLAN Map allowed tagging:11,13
```

NOTE: Configures a VLAN for a host and IOC pair. Related commands: vlanGet, vlanRem.

vlanRem

Remove the VLAN configuration

Syntax:

```
vlanRem <type> <host> <ioc> [<instance(s)>]
```

Options:

type
name|guid

host
host name/guid

ioc
IOC number

instance
host interface instance (optional)

Sample Output:

```
-> vlanRem name st130 1 0

vlanRem: action=2 target=4 defVlan=1 admitAll=1 IOC=1 PVID=1
hostInst=0




[Vi=81] [HostGUID=0x0002c90200216e18] [IOC=1] [Inst=0] [Name=st130]
---

ViPort: DefPVID=1 EgressFilter=no IngressFilter=no IngressFrame=all

VLAN Map Membership:--Empty--

VLAN Map allowed tagging:--Empty--

ExtPort[1]: DefPVID=1 EgressFilter=no IngressFilter=no
IngressFrame=all

VLAN Map Membership:--Empty--

VLAN Map allowed tagging:--Empty—
```

NOTE: Removes a VLAN configuration from a host and IOC pair. Related commands: vlanGet, vlanSet

ethExtInfoGet

Print information for the specified Ethernet port

Syntax:

```
ethExtInfoGet <port> or : ethExtInfoGet "<port1> <port2>
```

Options:

port

External Ethernet port to display information for.

Sample Output:

```
ethExtInfoGet 1
  External Ethernet port : 1
  Alias                  : Ext-1
  Enabled                : 1
  Mac Address            : 00:06:6a:00:45:50
  MTU Size               : 1518
  Default User Priority  : 2
  Pvid                   : 1
  STP Port State         : 0
  Acceptable Frame Types : 1
  Ingress Filtering      : 2
  Lcl Queues             : 0
  Lcl Queue Idx          : 0
  Ag Idx                 : 0
  Vlan Map Idx           : 0
  TpHCPortInDiscards     : 0
  Span Port              : 0
  Cos Count              : 0
  Ep Idx                 : 0
  Switch Port Idx        : 1
  Operator Status        : 2
  Error No               : 0
  Ls Index               : 1
```

NOTE: Prints information for the specified external Ethernet port. The port parameter can be either a single port or a list of ports separated by spaces and enclosed in double quotes.

ethVirtInfoGet

Print information for the specified virtual port

Syntax:

```
Usage: ethVirtInfoGet <port> or ethVirtInfoGet "<port1>
<port2>
```

Options:

port

Virtual port(s) to display information for.

Sample Output:

```
ethVirtInfoGet 1
  Virtual Nic            : 1
  Operator Status        : 2
  Error No               : 0
  External Ethernet port : 6
  Remote Host Types      : 0
  MTU Size               : 1
  Pvid                   : 1
  Default User Priority  : 0
  Acceptable Frame Types : 1
  Ingress Filtering      : 2
  Vi L2r Paths           : 0
  Vi L2r Path Index      : 0
  Ag Index               : 0
  Vlan Map Index         : 0
  TpHCPortInDiscards     : 0
  Span Port              : 0
  Vi Port Index          : 1
  Cos Count              : 0
  In Use                 : 1
  Last Host Name         : st131
  Last Host GUID         : 2249112375
  Last Host Instance     : 0
  Date Last Used         : WED DEC 13 13:28:08 2006

  Ioc Number             : 6
  Queue Pair Number      : 1078
  Switch Port Index      : 7
  MAC Address            : 00:06:6a:00:45:5c
  Multicast All Enable   : 0
  Promiscuous Enable     : 0
```

NOTE: Prints information for the specified virtual port. The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

ethExtInfoTable

Displays information for Ethernet ports in tabular format.

Syntax:

```
ethExtInfoTable type
```

Options:

type

0-summary|1-vlan

Sample Output:

```
-------------------------------------------------------------------------
EP|Up| MTU|PVID|Pri|Sts| SwP|       MAC       |Ang|DefSpd|CurSpd|
Alias
-------------------------------------------------------------------------
 1  3 1518    1   2   2    1 00:06:6a:00:b6:42   * 1G   f 1G   f    Ext-1
 2  3 1518    1   2   2    2 00:06:6a:00:b6:43   * 1G   f 1G   f    Ext-2
-------------------------------------------------------------------------
Enbl: "*"=> enabled, "-"=> disabled
Speed: "f"=> full duplex, "h"=> half duplex, "?"=> unknown
```

NOTE: Prints information for Ethernet ports.

ethVirtInfoTable

Displays information for the virtual ports in tabular format.

Syntax:

```
ethVirtInfoTable type
```

Options:

type

0-summary|1-vlan|2-host

Sample Output:

```
-------------------------------------------------------------------------VI  |
Swp|VNIC|Use| LastHostName   |Inst|Typ| MTU|      MAC        |IOC|Features
------------------------------------------------------------------------ 1    7
1   -            st11    0   0    0 00:06:6a:00:b6:4e   1 0001f0ff
   2    8    2   -            st11    1   0    0 00:06:6a:00:b6:4f   1 0001f0ff
   3    9    3   -            st11    2   0    0 00:06:6a:00:b6:50   1 0001f0ff
   4   10    4   -            st11    3   0    0 00:06:6a:00:b6:51   1 0001f0ff
   5   11    5   -            st11    4   0    0 00:06:6a:00:b6:52   1 0001f0ff
   6   12    6   -            st13    0   0    0 00:06:6a:00:b6:53   1 0001f0ff
   7   13    7   -            st13    1   0    0 00:06:6a:00:b6:54   1 0001f0ff
   8   14    8   -            st13    2   0    0 00:06:6a:00:b6:55   1 0001f0ff
   9   15    9   -            st13    3   0    0 00:06:6a:00:b6:56   1 0001f0ff
```

NOTE: Prints information for virtual ports.

ethExtMirrorGet

Print mirror information for the specified Ethernet port

Syntax:

```
ethExtMirrorGet <port> or ethExtMirrorGet "<port1>
<port2> , -> ethExtMirrorGet "1 2"
```

Options:

port

External Ethernet port (or list of ports) to retrieve mirror information for.

Sample Output:

```
External Ethernet port 1 [Ext-1] is not participating in a mirror.
External Ethernet port 2 [Ext-2] is not participating in a mirror.
```

NOTE: Displays mirroring information for the specified external Ethernet port. The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

ethExtMirrorSet

Create a mirror between two Ethernet ports.

Syntax:

```
ethExtMirrorSet <monitorPort> <mirrorPort>
```

Options:

monitorPort

External Ethernet port to be monitored.

mirrorPort

External Ethernet port to be the mirror.

Sample Output:

```
ethExtMirrorSet 1 2
Successfully setup mirror between monitor port 1 [Ext-1] and mirror
port 2 [Ext-
2]
```

NOTE: Sets up a mirror between the specified monitor and mirror external Ethernet port. Neither port can be a part of an existing mirror or aggregration. Note that mirror configurations are not saved across a reboot of the EVIC.

ethExtMirrorRem

Destroy a mirror between two Ethernet ports.

Syntax:

```
ethExtMirrorSet 1 2
Successfully setup mirror between monitor port 1 [Ext-1]
and mirror port 2 [Ext-
2]
```

Sample Output:

```
ethExtMirrorRem "1 2"
Removed External Ethernet port 1 [Ext-1] from mirror.
External Ethernet port 2 [Ext-2] is not in a mirror, ignoring.
```

ethExtEnabledGet

Displays whether a specified Ethernet port is enabled or disabled.

Syntax:

```
ethExtEnabledGet [port]
```

Options:

port

Ethernet port(s) from which to get the enabled state.

Sample Output:

```
ethExtEnabledGet "1 2"
External Ethernet port 1 [Ext-1] is enabled
External Ethernet port 2 [Ext-2] is enabled
```

NOTE: Displays the enabled state for the external Ethernet port. The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

ethExtEnable

Enable an Ethernet port(s).

Syntax:

```
ethExtEnable [port]
```

Options:

port
The Ethernet port(s) to enable.

Sample Output:

```
-> ethExtEnable 1
Enabled External Ethernet port 1 [Ext-1].
```

NOTE: Enables the external Ethernet port. The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

ethExtDisable

Disables the specified Ethernet port(s).

Syntax:

```
ethExtDisable [port]
```

Options:

port
The Ethernet port(s) to disable.

Sample Output:

```
-> ethExtDisable 1
Disabled External Ethernet port 1 [Ext-1].
```

NOTE: Disables the external Ethernet port. The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

ethExtCurrentSpeedGet

Display the speed/duplex mode currently in use by the specified Ethernet port.

Syntax:

```
ethExtCurrentSpeedGet <port> or ethExtCurrentSpeedGet
"<port1> <port2> ..."
```

Options:

port

External Ethernet port to read the speed/duplex mode currently in use.

NOTE: Displays the current speed/duplex values in use by the specified external Ethernet port. The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

ethExtLacpInfo

Show Lacp-related information on one or all external Ethernet ports.

Options:

port

Optional port number.

Sample Output:

```
ethExtLacpInfo [port]
```

NOTE: With no parameters, this command displays an overview of Lacp information for all external Ethernet ports The table contains the following information:

Port: The external Ethernet port that this line refers to. Link: Link status for the port (i.e. up or down).

Lacp: Indicates whether 802.3ad is enabled or disabled.

Agg#: 802.3ad or manual aggregation index.

State: Indicates success/failure of 802.3ad to put port into an aggregation

Failover: Minimum number of ports for failover to run (if link down).

Key: 802.3ad actor key for the port.

ID: 802.3ad actor system Id for the port.

PartnerPort: 802.3ad partner port number.

PartnerKey: 802.3ad partner key.

PartnerID: 802.3ad partner system Id.

For a manually configured aggregation, the following columns are displayed:

Port: The external Ethernet port that this line refers to.

Link: Link status for the port (i.e. up or down)

Agg#: Aggregation the port currently belongs to

Manual Agg #: Requested manual aggregation index

Failover: Minimum number of ports for failover to run (if link is down).

This command can optionally take an external Ethernet port number as it is the only arguement. In this case, the output will be detailed information about the link aggregation status and configuration for the specified port.

ethExtLacpEnable

Enables 802.3ad for the specified Ethernet port.

Syntax:

```
Usage: ethExtLacpEnable <port> or ethExtLacpEnable
"<port1> <port2> ..."
```

Options:

port

External Ethernet port (or list of ports) to enable Lacp.

Sample Output:

```
-> ethExtLacpEnable 1
802.3ad Lacp enabled for External Ethernet port 1 [Ext-1].
```

NOTE: Enables 802.3ad Lacp for the specified ports. If any of the ports are part of a manual aggregation, they will first be removed from the aggregation. The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

ethExtLacpDisable

Disable 802.3ad for the specified Ethernet port.

Options:

port
External Ethernet port (or list of ports) to disable Lacp.

Sample Output:

```
thExtLacpDisable 1
802.3ad Lacp disabled for External Ethernet port 1 [Ext-1].
```

NOTE: Disable 802.3ad Lacp for the specified port(s). If any of the ports are part of an aggregation, they will be removed before disabling Lacp. The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

ethExtLacpGet

Display enabled status of LACP.

Syntax:

```
ethExtLacpGet port
```

Options:

port
External Ethernet port (or list of ports) to display Lacp enabled information.

Sample Output:

```
-> ethExtLacpGet 1
802.3ad Lacp for External Ethernet port 1 [Ext-1]: enabled
-> ethExtLacpFailoverGet
 Port          Min Ports for Failover
--------------------------------------
  1 [Ext-1 ] : 0
  2 [Ext-2 ] : 0
```

NOTE: 802.3ad Lacp can be enabled or disabled on a per port basis. This command displays information on the Lacp enabled state for all specified Ethernet ports. The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

ethExtLacpKeyGet

Read the 802.3ad Lacp actor key for the specified port.

Syntax:

```
Usage: ethExtLacpKeyGet <port> or ethExtLacpKeyGet
"<port1> <port2> ..."
```

Options:

port

Displays the current Lacp actor key for the specified External Ethernet port.

Sample Output:

```
-> ethExtLacpKeyGet "1 2"
802.3ad Lacp actor key for External Ethernet port 1 [Ext-1]: 1
(00000001)
802.3ad Lacp actor key for External Ethernet port 2 [Ext-2]: 1
(00000001)
```

NOTE: 802.3ad allows for ports to be configured with an actor key, designating which ports can be a part of the same aggregation (i.e., only ports with the same actor key can be a part of the same aggregation, regardless of how the partner's information matches up). This command displays the Lacp key for the specified port(s). The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

ethExtLacpKeySet

Change the Lacp actor key for the specified External Ethernet port.

Syntax:

```
ethExtLacpKeySet <port> <key>
```

Options:

port

External Ethernet port (or list of ports) to modify the Lacp actor key.

key

New 16-bit Lacp actor key to assign to the port(s).

Sample Output:

```
-> ethExtLacpKeySet 1 2
Set 802.3ad Lacp actor key to 2 for External Ethernet port 1 [Ext-1].
```

NOTE: 802.3ad allows for ports to be configured with an actor key, designating which ports can be apart of the same aggregation (i.e., only ports with the same actor key can be a part of the same aggregation, regardless of how the partner's information matches up). This command modifies the Lacp key for the specified port(s). The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

ethExtAggregationGet

Display the current aggregation information for the specified port(s).

Syntax:

```
ethExtAggregationGet <port> or ethExtAggregationGet
"<port1> <port2> ..."
```

Options:

port

Display the current aggregation information for the External Ethernet port.

Sample Output:

```
-> ethExtAggregationGet "1 2"
Port    Alias     Method     Agg#
  1     Ext-1      disabled     0
  2     Ext-2      Lacp         0
```

NOTE: Link aggregation can be configured on a per-port basis to be determined automatically (via 802.3ad Lacp) or manually configured. This command prints the current aggregation status for the specified ports. The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

ethExtAggregationSet

Configure the specified External Ethernet port into the requested aggregation.

Syntax:

```
ethExtAggregationSet <port> <aggIdx> or
ethExtAggregationSet "<port1> <port2> ..." <aggIdx>
```

Options:

port

External Ethernet port (or list of ports) to configure into a manual aggregation.

aggIdx

Index/identifier for the manual aggregation to configure the port(s) into.

Sample Output:

```
-> ethExtAggregationSet 1 1
External Ethernet port 1 [Ext-1] manually configured into aggregation 1
```

NOTE: Link aggregation can be configured on a per-port basis to be determined automatically (via 802.3ad Lacp) or manually configured. This command allows the user to setup the port(s) into manually configured aggregations. The aggregation index is an arbitrary, non-zero index, and must refer to either an aggregation not currently in use, or one that has already been configured manually. Since manually configured configurations do not use Lacp, the user will also need to physically verify that the ports used in the aggregation are connected to the same switch, and that the switch is also manually configured to have those ports in an aggregation. Use aggregation index 0 to remove the port(s) from their respective manually-configured aggregations (note this does not re-enable Lacp for the ports). The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

ethExtLacpFailoverGet

Display the minimum number of active ports required for Link Aggregation failover.

Syntax:

```
ethExtLacpFailoverGet port
```

Options:

port

Port, list of ports, or '-all', indicating which ports to display.

Sample Output:

```
-> ethExtLacpFailoverGet

 Port          Min Ports for Failover

---------------------------------------

  1 [Ext-1 ] : 0

  2 [Ext-2 ] : 0
```

NOTE: When an external Ethernet link is removed or otherwise goes down, aggregation failover can take over and distribute the Ethernet traffic through other ports in the aggregation. This setting controls the minimum number of ports that must be remaining in an aggregation for aggregation failover to run on the specified port. If less than this number of ports are present, aggregation failover will not be enabled and any remote hosts using that port will see the link status as down. Aggregation failover is only relevant for ports in a manual or 802.3ad aggregation. The default setting is '0', which disables aggregation failover.

ethExtLacpFailoverSet

Modify the number of active ports required for Link Aggregation failover

Syntax:

```
ethExtLacpFailoverSet port min_ports
```

Options:

port

Port (or list of ports) to modify.

num_ports

Minimum number of ports necessary for failover.

Sample Output:

```
-> ethExtLacpFailoverSet 1 2

External Ethernet port 1 [Ext-1]: minimum number of ports for failover
set to 2
```

NOTE: When an external Ethernet link is removed or otherwise goes down, aggregation failover can take over and distribute the Ethernet traffic through other ports in the aggregation. This setting controls the minimum number of ports that must be remaining in an aggregation for aggregation failover to run on the specified port. If less than this number of ports are present, aggregation failover will not be enabled and any remote hosts using that port will see the link status as down. Aggregation failover is only relevant for ports in a manual or 802.3ad aggregation. The default setting is '0', which disables aggregation failover.

ethVirtAssignedHostGet

View assignment of virtual port.

Syntax:

```
ethVirtAssignedHostGet <instance(s)>
```

Options:

instance
port number

Sample Output:

```
-> ethVirtAssignedHostGet 61



ethVirtAssignedHostGet(0):

-----------------------------------------

Port  Host(2)               Inst IOC

  61  (none)                   0 0
```

NOTE: Show assignment of virtual port to attach to host. If an instance is not specified, all ports will be displayed. Related commands: ethVirtAssignedHostSet, ethVirtAssignedHostRem

ethVirtAssignedHostSet

Set assignment of the virtual port.

Syntax:

```
ethVirtAssignedHostSet <instance> <type> <host>
<instance> <ioc>
```

Options:

instance
port number

type
type=name|guid

host
host name/GUID

instance
host instance

ioc
IOC number

Sample Output:

```
ethVirtAssignedHostSet 1 name st61 3 1



ethVirtAssignedHostSet(1):

-----------------------------------------

Port  Host(4)               Inst IOC

   1  st61                     3 1

-----------------------------------------
```

NOTE: Create or update the assignment of the virtual port to attach to a host.
Related commands: ethVirtAssignedHostGet, ethVirtAssignedHostRem

ethVirtAssignedHostRem

Remove assignment of the virtual port.

Syntax:

```
ethVirtAssignedHostRem <instance(s)>
```

Options:

instance

port number

Sample Output:

```
ethVirtAssignedHostRem(2):
-----------------------------------------
Port  Host(2)               Inst IOC
  84  (none)                   0 0
-----------------------------------------
```

NOTE: Remove the assignment of virtual port to attach to a host. Related commands: ethVirtAssignedHostGet, ethVirtAssignedHostSet

ethExtAliasGet

Display the alias currently assigned for the Ethernet port(s).

Syntax:

```
ethExtAliasGet port
```

Options:

port

External Ethernet port (or list of ports) to get the alias for (can also be -all to display all ports).

Sample Output:

```
ethExtAliasGet -all
 Port  Alias
------------------
   1   Ext-1
   2   Ext-2
```

NOTE: Displays the port alias for all specified Ethernet ports. The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

ethExtAliasSet

Modifies the alias for the specified Ethernet port(s).

Syntax:

```
ethExtAliasSet port alias
```

Options:

port

External Ethernet port to assign the alias.

alias

New alias to assign to the port

Sample Output:

```
ethExtAliasSet 1 Ext-2
Set Ethernet port 1 alias to 'Ext-2'
```

NOTE: Assigns the alias to the specified Ethernet port(s). The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

ethVirtIgnoreVlanGet

Allows the user to view the IGNORE VLAN flag for a host port(s).

Syntax:

```
ethVirtIgnoreVlanGet <port(s)>
```

Options:

port(s)

Host port number(s). The port option can be a single port, or a list of ports separated by spaces and enclosed in double quotes.

Sample Output:

```
-> ethVirtIgnoreVlanGet 1
Port  Ignore VLAN

  1   Yes
```

NOTE: Show IGNORE VLAN flag for a host port. The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

ethVirtIgnoreVlanSet

Allows the user to set the IGNORE VLAN flag for a host port(s).

Syntax:

```
ethVirtIgnoreVlanSet <port(s)>
```

Options:

port(s)

Host port number(s). The port option can be a single port, or a list of ports separated by spaces and enclosed in double quotes.

Sample Output:

```
-> ethVirtIgnoreVlanSet 1
Port  Ignore VLAN

  1   Yes
```

NOTE: Set IGNORE VLAN flag for a host port. The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

ethVirtIgnoreVlanRem

Allows the user to remove the IGNORE VLAN flag for a host port(s).

Syntax:

```
ethVirtIgnoreVlanRem <port(s)>
```

Options:

port(s)

Host port number(s). The port option can be a single port, or a list of ports separated by spaces and enclosed in double quotes.

Sample Output:

```
-> ethVirtIgnoreVlanRem 1
Port  Ignore VLAN

  1   No
```

NOTE: Remove IGNORE VLAN flag from a host port. The port parameter can be either a single port, or a list of ports separated by spaces and enclosed in double quotes.

ethVirtVnic2McastGet

Reads if the Vnic2Mcast feature is enabled for a virtual port.

Syntax:

```
ethVirtVnic2McastGet [portNumber]
```

Options:

portNumber

The port to get enabled information for.

Sample Output:

```
-> ethVirtVnic2McastGet "1 2 3"
Port    Vnic2 Multicast
1       No
2       No
3       No
```

ethVirtVnic2McastSet

Enables the Vnic2Mcast feature for a virtual port.

Syntax:

```
ethVirtVnic2McastSet [portNumber]
```

Options:

portNumber

The port that has the Vnic2Mcast feature enabled..

Sample Output:

```
-> ethVirtVnic2McastSet "1 2 3"
Port    Vnic2 Multicast
1       Yes
2       Yes
3       Yes
```

NOTE: A reboot is required to activate changes.

ethVirtVnic2McastRem

Removes the Vnic2Mcast feature for a virtual port.

Syntax:

```
ethVirtVnic2McastRem [portNumber]
```

Options:

portNumber

The port to have the Vnic2Mcast feature removed.

Sample Output:

```
-> ethVirtVnic2McastRem 1
Port    Vnic2 Multicast
1       No
```

NOTE: A reboot is required to activate changes.

ethVnic2McastMtuGet

Reads the current Vnic2Mcast MTU size.

Syntax:

```
ethVnic2McastMtuGet
```

Options:

None.

Sample Output:

```
-> ethVnic2McastMtuGet
Link MTU for VNIC2 multicast groups : 4 (=> 2048)
```

ethVnic2McastMtuSet

Modifies the current Vnic2Mcast MTU size.

Syntax:

```
ethVnic2McastMtuSet [Vnic2McastMtu size]
```

Options:

Vnic2McastMtu size

The Vnic2McastMtu size. Type the command with no size to display options.

Sample Output:

```
-> ethVnic2McastMtuSet
usage: ethVnic2McastMtuSet <mtu>
mtu = 1 (256), 2 (512), 3 (1024), 4 (2048), 5 (4096),

-> ethVnic2McastMtuSet 4
Successfully set link MTU (4) for VNIC2 Multicast groups.
Link MTU for VNIC2 multicast groups : 4 (=> 2048)
```

NOTE: Changes take effect following a reboot. The Vnic2McastMTU size must match the MTU size of the IB fabric.

ethVnic2McastRateGet

Reads the current Vnic2Mcast rate setting used during registration.

Syntax:

```
ethVnic2McastRateGet
```

Options:

None.

Sample Output:

```
-> ethVnic2McastRateGet
Link rate for VNIC2 multicast groups : 3 (=> 4x or 10G)
```

ethVnic2McastRateSet

Modifies the Vnic2Mcast rate setting used during registration.

Syntax:

```
ethVnic2McastRateSet <Vnic2Mcast rate>
```

Options:

Vnic2Mcast rate

The Vnic2Mcast rate setting to use. Type the command with no rate to display options.

Sample Output:

```
-> ethVnic2McastRateSet
usage: ethVnic2McastRateSet <rate>
rate = 0 (Don't care), 1 (1G), 2 (1x or 2.5G), 3 (4x or 10G), 4 (12x or
30G), 5 (5G), 6 (20G), 7 (40G), 8 (60G), 9 (80G), 10 (120G),

-> ethVnic2McastRateSet 3
Successfully set link rate (3) for VNIC2 Multicast groups.
Link rate for VNIC2 multicast groups : 3 (=> 4x or 10G)
```

NOTE: A reboot is required to activate changes.

Configuration

confFileSave

Saves the current configuration to a file on the local ramdisk in the `/config` directory.

Syntax:

```
confFileSave <file>
```

Options:

file
Name of the file in the /config directory to save the current configuration to.

Sample Output:

```
-> confFileSave
```

NOTE: This routine generates a text-based configuration file from the current configuration. It is saved into the `/config` directory. Note that the `/config` directory is volatile and all contents will be lost on a reboot. It is expected that the user will copy the configuration file off the device using FTP.

confFileCheck

Validates a configuration file.

Syntax:

```
confFileCheck <filename>
```

Options:

filename
Name of the configuration file to validate.

Sample Output:

```
-> confFileCheck
```

NOTE: Loads and validates a configuration file, ensuring it's syntax and settings at least appear valid.

confFileCommit

Loads and commits a configuration file to memory.

Syntax:

```
confFileCommit <filename>
```

Options:

filename
Name of the configuration file to commit.

Sample Output:

```
-> confFileCommit
```

NOTE: Loads, validates, and applies a configuration file to the system.

confFileList

List contents of the `/config` directory.

Syntax:

```
confFileList
```

Options:

None.

Sample Output:

```
-> confFileList
```

NOTE: List the contents of the `/config` directory. Note that the `/config` directory is volatile; all contents will be lost on a reboot.

confFileShow

Displays contents of a configuration file

Syntax:

```
confFileShow <filename>
```

Options:

filename
The name of the file to display

Sample Output:

```
-> confFileShow
```

NOTE: Displays the contents of a configuration file to the console.

confFileBackup

Transfer a saved configuration file to a remote FTP server.

Syntax:

```
confFileBackup <host> <user> <pass> <dir> <file>
```

Options:

host
The IP address of the FTP server.

user
User name for login to the FTP server.

pass
Password for login to the FTP server.

dir
Remote directory location to store the file.

file
File name of the configuration file to be transferred.

Sample Output:

```
-> confFileBackup
```

NOTE: Transfers a saved configuration file from the card to a remote FTP server. The filename parameter is used as both the local filename, and the filename to save the file as on the remote server.

confFileRetrieve

Retrieve a configuration file from an FTP server.

Syntax:

```
confFileRetrieve <host> <user> <pass> <dir> <file>
```

Options:

host
The IP address of the FTP server.

user
User name for login to the FTP server.

pass
Password for login to the FTP server.

dir
Remote directory location to retrieve the file from.

file
File name of the configuration file to be retrieved.

Sample Output:

```
-> confFileRetrieve
```

NOTE: Transfers a configuration file from the remote FTP server to the local configuration directory. No validation of the configuration file is performed. You will need to manually validate and then commit the config file for it's settings to take effect.